Short-Term Power Output Forecasting for Large Multi-Megawatt Photovoltaic Systems with an Aggregated Low-Level Forecasting Methodology

by

Armand A. du Plessis

Dissertation presented for the degree of Doctor of Philosophy in the Faculty of Engineering at Stellenbosch University.

Supervisors: Dr. Arnold J. Rix
Dr. Johannes M. Strauss

March 2021

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 2020/11/4

Abstract

Short-Term Power Output Forecasting for Large Multi-Megawatt Photovoltaic Systems with an Aggregated Low-Level Forecasting Methodology

A. A. du Plessis

Department of Electrical and Electronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Dissertation: Presented for the degree of Doctor of Philosophy in the Faculty of Engineering at Stellenbosch University

March 2021

Power delivered from utility-scale Photovoltaic (PV) systems is characteristically intermittent, due to a dependence on atmospheric variables. To manage this uncertainty of an intermittent PV power supply, researchers traditionally adopt a macro-level forecasting approach, where a single model is trained to emulate the behaviour of the entire PV system. However, as commercial PV systems continue to expand in size, there is a growing uncertainty regarding the ability of these macro-level models to capture the non-uniform, low-level power output dynamics of large multi-megawatt PV systems. In response to this knowledge gap, a novel aggregated low-level forecasting methodology is proposed.

With state-of-the-art deep learning (DL) implementations of Feedforward Neural Network (FFNN), Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) and Gated Recurrent Unit Recurrent Neural Network (GRU-RNN) models, the proposed methodology is compared to the conventional macro-level forecasting approach. With data obtained from a commercial 75 MW PV system, multi-step 1 - 6 h ahead forecasts are delivered for a real-world scenario. Forecast models are trained for each of the 84 inverters, which collectively serve as the aggregated low-level forecasting solution. However, given the high computational expense of training multiple forecast models, a unique and scalable inverter-clustering approach towards model development is presented. The discrepancies in literature concerning biased model development are also addressed, with a heuristic process of systematic hyper-parameter optimisation proposed, which serves to guide future forecasting practitioners.

Concerning the results, this research successfully demonstrates the application of the proposed methodology. From the day-time-only forecast results, the aggregated inverter-level FFNN model shows the largest improvement, with a Mean Absolute Percentage Error (MAPE) of between 0.04 % - 0.4 % lower in comparison to the FFNN macro-level forecasts. This translates to an overall 30 kW - 300 kW improvement in forecasting accuracy. The aggregated GRU-RNN inverter-level model forecasts deliver a smaller overall MAPE performance increase, ranging between 0.03 % - 0.1 %. This is a 20 kW - 75 kW improvement.

However, compared to all the DL forecast models applied, the low-level GRU-RNN model forecasts deliver the highest overall forecasting accuracy, with MAPE values ranging between 5.8 % - 8 % (day-time-only forecasts). From the 95 % Bootstrap confidence intervals, no improvements regarding the uncertainty analysis are observed for the aggregated low-level forecasting methodology. Finally, with this research it is concluded that researchers who have and continue to propose DL-based forecast model solutions for smaller multi-megawatt PV systems, can be confident in the application of these models as macro-level solutions.

Uittreksel

Korttermyn Drywingsuitsetvoorspelling van Groot Multi-Megawatt Fotovoltaïese Stelsels met 'n Saamgevoegde Laevlak Voorspellings Metodologie

("Short-Term Power Output Forecasting for Large Multi-Megawatt Photovoltaic Systems with an Aggregated Low-Level Forecasting Methodology")

A. A. du Plessis

Departement Eletriese en Elektroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Proefskrif: Proefskrif ingelewer vir die graad Doktor in Filosofie in die Fakulteit Ingenieurswese aan die Universiteit Stellenbosch

Maart 2021

Die konvensionele benadering om voorspellingsmodelle te ontwikkel, wat die drywingsuitset van groot multi-megawatt Fotovoltaïese (FV) kragstelsels voorspel, word baseer op 'n makrovlakvoorspellingsmetodologie, waar 'n enkele model geleer word om die gedrag van die hele FV-stelsel te voorspel. Soos wat kommersiële FV-stelsels uitbrei in grootte is daar egter 'n toenemende onsekerheid rakende die vermoë van hierdie makrovlakmodelle om die nie-uniforme, lae-vlak drywingsuittreedinamika vas te vang van hierdie groot FV-stelsels. In antwoord op hierdie kennisgaping word 'n nuwe FV-voorspellingsmetodologie voorgestel, waar dit ondersoek word of 'n verbeterde voorspellingsakkuraatheid bereik kan word met 'n samevoeging van veelvuldige lae-vlak voorspellingsmodelle. Dit is belangrik, omdat verdere verbetering wat in voorspellingsakkuraatheid gemaak word krities is vir elektrisiteitsnetwerkoperateurs, wie die toegevoegde druk van 'n wisselvallige FV-stelsel as energiebron effektief moet bestuur.

Om die fokus van die navorsing op die voorgestelde metodologie te behou, word geen hibriede modelle oorweeg nie, met slegs alleenstaande, mees gevorderde Diep-Leer (DL) modelle, wat insluit 'n Voer-Vorentoe-Neurale-Netwerke (VVNN), Lang-Korttermyngeheue-Herhalende-Neurale-Netwerke (LKTGHNN) en Hek-Herhalende-Eenheid-Herhalende-Neurale-Netwerke (HHEHNN), wat toegepas word. Met hierdie DL-modelle word multistap voorspellings 1 - 6 h vooruit gelewer vir 'n 75 MW netwerkgeskakelde FV-stelsel. Voorspellingsmodelle is vir elkeen van die 84 wisselrigters van die FV-stelsel ontwikkel, wat sodoende dien as die saamgevoegde laevlak voorspellingsoplossing. Die berekeningsuitdaging aangaande die skaleerbaarheid en reproduseerbaarheid van die verskeie DL-gebasseerde modelle is ook suksesvol aangespreek met 'n unieke wisselrigter-groeperingstegniek. Verder, as antwoord op die geïdentifiseerde ongelykheid in navorsing in die literatuur, rakende die onregverdige en

onoortuigende aanspraak van model meerderwaardigheid, word 'n heuristiese proses van stelselmatige hiperparameter-optimalisering voorgestel, wat dien as riglyn tot onbevooroordeelde model ontwikkeling vir die FV-voorspellingspraktisyn.

Hierdie navorsing demonstreer die toepassing van die voorgestelde metodologie suksesvol. Die resultate van die daglig-alleen-voorspellings wys dat die saamgevoegde wisselrigtervlak-VVNN-model die grootste verbetering toon, met 'n Gemiddelde-Absolute-Persentasie-Fout (GAPF) van tussen 0.04 % - 0.4 % laer in vergelyking met makrovlak VVNN-voorspellingsmodel. Dit kom neer op 'n algehele 30 kW - 300 kW verbetering in voorspellingsakkuraatheid. Die saamgevoegde wisselrigtervlak-HHEHNN-model lewer egter 'n kleiner algehele GAPF-prestasie-verbetering, wat wissel tussen 0.03 % - 0.1 %. Dit is 'n 20 kW - 75 kW verbetering. Nietemin, in vergelyking met al die toegepaste DL-voorspellingsmodelle, het die saamgevoegde wisselrigtervlak-HHEHNN-model die hoogste algehele voorspellingsakkuraatheid bereik, met GAPF waardes wat wissel tussen 5.8 % - 8 % (daglig-alleen-voorspellings). Met die 95 % Bootstrap-sekerheidsinterval is daar geen verbeteringe aangaande die onsekerheidsanalise gevind vir die saamgevoegde laevlak-voorspellingsmetodologie nie. Laastens, met hierdie navorsing is daar tot die gevolgtrekking gekom dat navorsers, wie alreeds in die verlede, of nog beplan om nuwe DL-modelle te ondersoek as voorspellingsoplossings vir kleiner multi-megawatt FV-stelsels, met selfvertroue die toepassing van hierdie modelle as makrovlak-oplossings kan aanwend.

Acknowledgements

God, for entrusting me with this wonderful opportunity and for helping me make a success of this dissertation. Your guidance, goodness, inspiration and support has been unwavering during this journey, allowing me to do exceedingly more than I thought possible. Your blessing has been apparent in every step of this project and has allowed each effort to be multiplied in outcome. I am fortunate to have and to know You as my Father and I am looking forward to our next journey together!

I would like to express my sincere gratitude to the following people...

In no particular order, to my two study leaders Dr Johannes M. Strauss and Dr Arnold J. Rix. The three of us have walked a long road together with this project. Thank you for mentoring me and for guiding my growth as an engineer. You have taught me to conceptualise and think both critically and philosophically about solving problems. I hope that you will not for one moment underestimate the impact that you have made on my life. Although our main focus was this dissertation, it feels like the sum of the parts is so much more. I am truly privileged to know you and I deeply cherish our friendship formed over these years.

My parents, André and Henriette du Plessis who have supported me on this journey beyond the call of duty. One simply cannot fail at life with parents like you. Thank you for the example you have set and for showing me the value of hard work. Even greater than this, for drenching me with an abundance of love, support and guidance.

To everyone in the Media Lab, which has served as my home during these few years. In particular, a special thank you to Kurt Coetzer, Warren Farmer, Leanne Nortje, Francisca Daniels, Ian Reischauer, Daniël Schoonwinkel, Brian De Beer, Jobert Louw, Tafadzwa Gurupira, Seyi Paul Babalola, Carel Landman, Elan Van Biljon, Kayode Olaleye, Benjamin van Niekerk, Burger Becker, Kevin Buresh, Jason Samuels, Jeanne Daniels, Ryan Eloff, Meli Menelaos, Daniel Robinson, Innocent Ndibatya, Hermé Smit, Lisa van Staden, Amaris Dalton, Oan De Waal, James Ellis, Daniel De Villiers and Lewis Waswa. Also, a big thank you to you Arnu Pretorius who has been a great friend and a mentor to me.

Wayne Dyamond, for pushing through many hours with me to clean our data set and for the occasional conversation of course. You certainly played an important part in the success of this research and for that I am eternally grateful.

Dr Herman Kamper, for your much appreciated guidance and eagerness to help me during my journey of mastering machine learning.

To my family at Grace Life, who have tirelessly supported me in this endeavour.

To my new family. Marcel & Sureta De Ridder, you have truly made me a son in your home and you have tremendously supported me during this endeavour. Thank you for all the meals and conversations and for being cheerleaders along the way. Also, to Marcell De Ridder, Ryno De Ridder, Leandii De Ridder and Kylé Noome. Your friendship and support during this project has been awesome.

And finally my wife, Terusa du Plessis. You have stood with me every single step of this adventure. Your overwhelming supply of love, support, understanding and deeply appreciated friendship has truly been in abundance. Thank you for tirelessly being there over many weekends, late nights and even entire holidays. Your sacrifices for the greater good, which was this dissertation, has been equal and perhaps greater to some extent than mine. Without you it would not have been the amazing journey that it was. This was truly a team effort. I am forever and always grateful to you. I just love you so much.

Dedications

This dissertation is dedicated to Terusa...

Contents

Declaration i

Abstract ii

Uittreksel iv

Dedications viii

Contents ix

List of Figures xiii

List of Tables xvii

Nomenclature xx

1 Introduction 1
- 1.1 The need for solar power forecasting . . . 1
- 1.2 Current state of short-term PV forecasting . . . 2
  - 1.2.1 Forecasting fundamentals . . . 2
  - 1.2.2 Forecast horison . . . 3
  - 1.2.3 Model input data sources . . . 3
  - 1.2.4 Performance metrics . . . 4
  - 1.2.5 PV-forecasting model selection . . . 6
  - 1.2.6 Artificial intelligence techniques . . . 8
  - 1.2.7 Utility-scale PV-system forecasting . . . 18
  - 1.2.8 Critique and summary of PV-forecasting research . . . 18
- 1.3 Research motivation . . . 19
- 1.4 Research aims . . . 22
- 1.5 Research hypothesis . . . 23
- 1.6 Research objectives . . . 23
- 1.7 Research scope and limitations . . . 23
  - 1.7.1 Forecast horison . . . 23
  - 1.7.2 Input and output data . . . 24
  - 1.7.3 Model selection . . . 24
  - 1.7.4 Single-target vs. multi-target regression . . . 25
  - 1.7.5 Day-type and seasonal forecasting . . . 25
- 1.8 Thesis overview . . . 26

2 Data Processing 27
- 2.1 Introduction and overview . . . 27
- 2.2 PV-system location and environment . . . 27

2.3 Overview of data processing strategy . . . 29
2.4 Data pre-processing Stage 1: Raw data acquisition . . . 29
2.4.1 Weather sensor data . . . 29
2.4.2 PV-system data . . . 30
2.4.3 Data acquisition and storage . . . 33
2.5 Data pre-processing stage 2: Raw data clean up . . . 33
2.5.1 Bad data elimination . . . 33
2.5.2 Data interpolation . . . 34
2.5.3 Data imputation and outlier elimination . . . 34
2.6 Data pre-processing stage 3: Feature engineering and data preparation . . . 35
2.6.1 Feature engineering . . . 35
2.6.2 Data preparation: Training, validation and test sets . . . 38

3 Forecast model development 40
3.1 Overview . . . 40
3.2 Feedforward Neural Networks . . . 40
3.2.1 Fundamental operation . . . 40
3.2.2 Activation function . . . 42
3.2.3 FFNN model training and cost function . . . 42
3.3 Recurrent Neural Networks . . . 44
3.3.1 Input data processing . . . 44
3.3.2 RNN layer and training . . . 45
3.3.3 RNN training . . . 46
3.3.4 Long Short-Term Memory Recurrent Neural Network . . . 47
3.3.5 Gated Recurrent Unit Recurrent Neural Network . . . 49
3.4 Input feature selection . . . 51
3.4.1 Sliding window . . . 52
3.4.2 Feature scaling . . . 56
3.5 Model training and fine-tuning . . . 56
3.5.1 Optimiser, learning rate and mini-batch size . . . 56
3.5.2 Activation function . . . 57
3.5.3 Early-stopping and regularisation . . . 57
3.5.4 Data reproducibility . . . 57
3.6 Hyperparameter optimisation . . . 57
3.6.1 The issue of claimed model superiority . . . 58
3.6.2 PV-system macro-level forecast model development . . . 58
3.6.3 Inverter-level model development . . . 61

4 Results and discussion 69
4.1 Overview . . . 69
4.2 Evaluation metrics . . . 69
4.2.1 Deterministic . . . 69
4.2.2 Uncertainty analysis . . . 70
4.3 Macro-level PV-system forecasts . . . 72
4.3.1 Overview . . . 72
4.3.2 Phase-1 development . . . 72
4.3.3 Phase-2 and Phase-3 development . . . 75
4.3.4 Macro-level model development summary . . . 78
4.3.5 Results . . . 78
4.3.6 Contextualising model accuracies . . . 83

4.4 Aggregated inverter-level forecasts . . . 86
4.4.1 Model development framework . . . 86
4.4.2 FFNN model development . . . 88
4.4.3 FFNN: macro-level vs. inverter-level forecasts . . . 92
4.4.4 GRU-RNN model development . . . 96
4.4.5 GRU-RNN: macro-level vs. inverter-level forecasts . . . 99
4.5 Summary of results . . . 101
4.5.1 Direct comparison of FFNN and GRU-RNN models . . . 101
4.5.2 Real-world forecasting results . . . 105

5 Conclusions and recommendations 109
5.1 Research summary . . . 109
5.2 Conclusion . . . 111
5.3 Research contributions . . . 113
5.4 Recommendations . . . 113
5.5 Future research . . . 114

Appendices 115

A PV-system data 116
A.1 Available weather sensor data . . . 116
A.2 PV-system power output related data . . . 116
A.3 Data acquisition and storage . . . 118
A.4 Data processing . . . 120
A.4.1 Data resolution . . . 120
A.4.2 Data pre-processing: Interpolation . . . 120
A.4.3 Data pre-processing: Imputation . . . 123
A.4.4 Curtailment elimination . . . 141
A.4.5 Data pre-processing concluded . . . 142
A.5 Correlation analysis . . . 143
A.5.1 Pearson's correlation method . . . 143
A.5.2 Spearman's rank-correlation method . . . 144
A.5.3 Application of Pearson and Spearman correlation methods . . . 144

B Earth Sun movement 146
B.1 Sun position relative to a specific location . . . 146

C Macro-level model development 150
C.1 FFNN . . . 150
C.2 LSTM-RNN . . . 150
C.3 GRU-RNN . . . 151
C.4 Macro-level forecast model time-series graphs . . . 151

D Inverter-level model development 154
D.1 FFNN sliding-window preference . . . 154
D.2 FFNN inverter-cluster models: Models A and B . . . 155
D.3 GRU-RNN sliding window preference . . . 156
D.4 GRU-RNN inverter-cluster models: Models C and D . . . 158
D.5 GRU-RNN inverter-cluster models: Models E and F . . . 161

E Day-time-only forecast results 164

E.1 Forecast results 164

F Training time **168**
F.1 Forecast model training time 168

G Inverter clustering **169**
G.1 Defining the number of clusters 169

List of References **173**

List of Figures

1.1 Relationship between factors that define the framework of PV-forecasting model development. . . . 2
1.2 Classification of the various PV forecasting techniques. . . . 7
1.3 Classification of artificial intelligence techniques. . . . 8
1.4 A conventional macro-level PV forecasting approach. . . . 20
1.5 Aerial view of the 75 MW PV-system used for the execution of this research. . . . 20
1.6 Demonstration of the non-uniform effects of wind on PV-system power output dynamics. . . . 21
1.7 Les Mées PV system with a non-uniform (rolling-hills) ground surface topology. 21
1.8 Suggested approach of aggregated low-level PV-system forecasts. . . . 22
1.9 Illustration of four different day types as determined by received irradiance. . . . 26

2.1 Aerial view from the side of the 75 MW PV-system used for the execution of this research . . . 27
2.2 Location of PV system used for this research and other operational and planned utility-scale PV systems in South Africa. . . . 28
2.3 Three-stage data processing pipeline. . . . 29
2.4 Weather station locations within the PV power plant. . . . 30
2.5 Low-level PV-system layout of the primary hardware together with important points of data capture. . . . 31
2.6 Ambient temperature boundary condition elimination. . . . 33
2.7 Example of interpolation applied to missing power data. . . . 34
2.8 Illustration of the sun position as determined by the altitude β and azimuth Φ_S angles. . . . 36
2.9 Illustration of annual and daily seasonality for the power output and GHI data. 36
2.10 Wind direction and speed measured. . . . 37
2.11 Partitioning of the training, validation and test data sets. . . . 38

3.1 Illustration of the basic ANN structure and neuron. . . . 41
3.2 FFNN input data. . . . 45
3.3 RNN input data. . . . 45
3.4 RNN cell unfolding. . . . 46
3.5 LSTM cell. . . . 47
3.6 GRU cell . . . 49
3.7 GRU-RNN model implementation. . . . 51
3.8 Example of multivariate sliding window approach applied for data preparation. . . . 53
3.9 Distinction between the inverter and macro PV-system data sets. . . . 54
3.10 Illustration of how data set samples are provided to GRU-RNN, LSTM-RNN and FFNN models. . . . 55
3.11 Illustration of a data segment with a clear trend. . . . 55
3.12 Macro-level forecast model development framework. . . . 58

3.13 Illustration of the relationship between the sliding windows, input features, hidden layers and hidden units. . . . 60
3.14 Example of a hyperparameter grid-search domain. . . . 60
3.15 Illustration of Phase-2 guided grid-search process. . . . 61
3.16 Inverter assignment to clusters. . . . 62
3.17 Euclidean distance similarity for all inverters. . . . 63
3.18 Division of inverter blocks into 10 clusters. . . . 65
3.19 Forecast model development framework for inverter models. . . . 65
3.20 Difference between summed inverter power and macro PV-system power output. 66
3.21 Aggregated inverter power output directly compared to measured macro-level PV system power output. . . . 67
3.22 Correction of aggregated inverter-level forecasts with IPL model. . . . 67
3.23 Total inverter power output validation set corrected. . . . 68

4.1 Example of the bootstrap process to obtain a 95% confidence interval. . . . 71
4.2 Macro-level forecast model development framework. . . . 72
4.3 Time-series forecasts of macro-level models for 1 h, 3 h and 6 h ahead. . . . 79
4.4 Test set NRMSE [%] results of macro-level forecast models as delivered for specific weather conditions. . . . 80
4.5 Test set MAPE [%] results of macro-level forecast models as delivered for specific weather conditions. . . . 81
4.6 Test set MAPE [%] bootstrap analysis of the macro-level forecast models for all weather conditions. . . . 81
4.7 Demonstration of a basic fully connected Linear regression FFNN model. . . . 83
4.8 Time-series forecasts of the developed FFNN, LSTM-RNN and GRU-RNN models compared to a Linear regression baseline model. . . . 84
4.9 Averaged test set MAPE performance for all weather conditions as delivered for each time step by the FFNN, LSTM-RNN, GRU-RNN and Linear regression models. . . . 85
4.10 Forecast model development framework for aggregated inverter-level models. . . . 86
4.11 Demonstration of the non-uniform effects of wind on the power output dynamics of the inverters. . . . 87
4.12 Inverter clusters with hyperparameter assignment visualised. . . . 88
4.13 Averaged FFNN model MAPE inverter-level results compared to a Linear regression baseline model. . . . 91
4.14 Forecast model approach comparison: PV-system macro-level vs. Inverter-cluster vs. aggregated individual inverter models. . . . 92
4.15 Test set time-series forecasts of FFNN PV-system macro-level and aggregated inverter-level models for 1 h, 3 h and 6 h ahead. . . . 93
4.16 Test set NRMSE [%] results of FFNN aggregated inverter-level and macro-level forecast models. . . . 94
4.17 FFNN aggregated inverter-level and macro-level PV-system MAPE [%] Bootstrap distributions for all weather types of the test data set. . . . 95
4.18 Averaged GRU-RNN model MAPE inverter-level results compared to a Linear regression baseline model. . . . 98
4.19 Test set time-series forecasts of GRU-RNN PV-system macro-level and aggregated inverter-level models for 1 h, 3 h and 6 h ahead. . . . 99
4.20 Test set NRMSE [%] results of GRU-RNN aggregated inverter-level and macro-level forecast models. . . . 100

4.21 GRU-RNN inverter-level and macro-level PV-system MAPE [%] Bootstrap distributions for all weather types. . . . 101
4.22 Inverter-level and macro-level test set MAPE [%] results for GRU-RNN and FFNN models. . . . 102
4.23 All test set MAPE Bootstrap CIs for the GRU-RNN and FFNN aggregated inverter-level and macro-level forecast accuracies. . . . 103
4.24 Relationship of FFNN and GRU-RNN forecasting accuracy differences. . . . 104
4.25 Difference between the inverter-level and macro-level MAPE results obtained, with positive difference values indicating the improved forecasting accuracy from the inverter-level models. . . . 107

5.1 Example of a large PV system constructed on complex terrain. . . . 114

A.1 QNAP NAS drive. . . . 119
A.2 Raw and clean data database structure. . . . 119
A.3 Deployment of database onto docker container and linking with ML models hosted on another Docker container. . . . 119
A.4 Data set resolution comparison. . . . 120
A.5 Example of linear and polynomial interpolation applied. . . . 121
A.6 Example of linear interpolation applied to data missing for a 1 h, 2 h and 3 h period. . . . 122
A.7 Example of linear interpolation strategy applied to missing data segments. . . . 123
A.8 Example of data elimination where one of the input features has more than 1 h of missing data. . . . 123
A.9 Measured temperature data from multiple weather sensors. . . . 125
A.10 Illustration of missing and a prolonged temperature data deviation from group average for weather sensor WS-TR15. . . . 125
A.11 Final averaged temperature sensor data set after all data deviations are eliminated. 125
A.12 Measured GHI data from multiple weather sensors. . . . 126
A.13 Illustration of constant irradiance data deviation from group average for weather sensor WS-CB. . . . 126
A.14 GHI data set differences relative to group average for all weather sensors after eliminating WS-CB weather sensor data. . . . 127
A.15 Final averaged GHI data set used for model training. . . . 127
A.16 Wind speed measurements of all weather stations. . . . 128
A.17 Wind speed differences relative to group average for all weather stations. . . . 128
A.18 New wind speed differences relative to group average for all weather stations after the elimination of WS-TR15 for the period of 2015-05 to 2018-01. . . . 128
A.19 Observed wind speed measurement deviation of WS-TR27 from group norm. . . . 129
A.20 Wind-speed differences from group norm, after the elimination of WS-TR27 data segment for 2015-5 to 2015-11. . . . 129
A.21 Final averaged wind speed data set. . . . 130
A.22 Wind-direction measurements. . . . 130
A.23 Absolute-air-pressure measurements. . . . 131
A.24 Relative-humidity measurements. . . . 131
A.25 Missing weather station data imputed. . . . 132
A.26 Distinction between the inverter data sets used for inverter-level models and the total PV-system power output data set used for macro-level forecast model development. . . . 132
A.27 Percentage of missing data for each inverter and the macro PV-system power data sets. . . . 133

A.28 Total consecutively missing data entries for each power source. 134
A.29 Top: Total maximum consecutive zero data entries for each power source Bottom: Total consecutive zero data entries counted for each power source without I63. . 134
A.30 Inverter MSE correlation matrix 136
A.31 PV-system layout with inverter positions 137
A.32 Power data adjustment made to both inverter and macro PV-system data sets. . 138
A.33 Power data adjustment made only to inverter data set. 138
A.34 Average daily power difference between measured aggregated inverter power and the total Zone-3 measurement. 139
A.35 Average daily power difference between measured aggregated inverter power and the total Zone-6 measurement. 140
A.36 Inverter I34 imputed data and macro PV-system power data set adjustment. . . 141
A.37 Occurrences of curtailment during the entire data set period. 142
A.38 Summary of the data set irregularities that have been addressed. 143
A.39 Real-time correlations of power relative to the various measured data points. . . 144
A.40 Spearman correlation analysis between historic data and real-time PV-system power output. 145

B.1 Illustration of the ecliptic movement around the Sun with Earth at a tilt angle. 146
B.2 Illustration of the resulting Declination angle as a result of the Earth and Sun movement. 147
B.3 Illustration of Sun position as determined by altitude angle β and azimuth angle Φ_S. 147

C.1 Time-series forecasts delivered 1 h, 3 h and 6 h ahead for clear days. 152
C.2 Time-series forecasts delivered 1 h, 3 h and 6 h ahead for clear-intermittent days. 152
C.3 Time-series forecasts delivered for 1 h, 3 h and 6 h ahead for intermittent days. 153
C.4 Time-series forecasts delivered for 1 h, 3 h and 6 h ahead for overcast days. . . . 153

E.1 Day-time-only test set MAPE Bootstrap CIs for the GRU-RNN and FFNN aggregated inverter-level and macro-level forecast accuracies. 165

G.1 Number of inverter-clusters to be allocated as indicated Elbow method. 169
G.2 Number of inverter-clusters to be allocated as indicated CH-index. 170
G.3 Number of inverter-clusters to be allocated as indicated with Gap statistic. . . . 170
G.4 Dunn index cluster evaluation metric results. 171
G.5 Davies-Bouldin index cluster evaluation metric results. 172

List of Tables

1.1 Forecasting horison definitions. 3
1.2 Spatial and temporal correlation of the available data sources. 5
1.3 Summary of the short-term forecasting literature results obtained with machine learning models as solution. 11
1.3 Summary of the short-term forecasting literature results obtained with machine learning models as solution. 12
1.3 Summary of the short-term forecasting literature results obtained with machine learning models as solution. 13
1.3 Summary of the short-term forecasting literature results obtained with machine learning models as solution. 14
1.4 Summary of the short-term forecasting literature results obtained with deep learning models as solution. 17

2.1 PV system weather data used for model development. 30
2.2 Available PV-system data sources related to power output. 30
2.3 PV-system power data points recorded and used for model development. 32
2.4 One-hot encoding of measured wind direction values. 37

3.1 Popular activation functions. 43
3.2 Final set of available input features. 54
3.3 Input feature and sliding window combinations. 59

4.1 Phase-1 initial grid-search solution space defined for each model. 73
4.2 Average validation set NRMSE results obtained for FFNN macro-level model forecasts as evaluated for specific input features and all weather conditions. . . . 74
4.3 Average validation set NRMSE results obtained for LSTM-RNN macro-level model forecasts as evaluated for specific input features and all weather conditions. . . . 74
4.4 Average validation set NRMSE results obtained for GRU-RNN macro-level model forecasts as evaluated for specific input features and all weather conditions. . . . 75
4.5 Averaged NRMSE results of best FFNN macro-level models for Phase-1 and Phase-2 model development. 76
4.6 Averaged NRMSE results of best LSTM-RNN macro-level models for Phase-1 and Phase-2 model development. 77
4.7 Averaged NRMSE results of best GRU-RNN macro-level models for Phase-1 and Phase-2 model development. 77
4.8 Best macro-level PV-system forecasting model details. 78
4.9 PV-system macro-level forecast results for all models and weather conditions as determined from the test set. 79
4.10 Test set 95 % Bootstrap CIs of the macro-level forecast models for all weather conditions. 82
4.11 Comparison of macro-level GRU-RNN model results to published literature. . . 85

4.12 FFNN aggregated inverter-level model development Phase-1 solution space. . . . 89
4.13 FFNN *A* inverter-level Phase-1 best HU and HL combinations. 90
4.14 FFNN *B* inverter-level Phase-1 best HU and HL combinations. 90
4.15 Validation-set results for aggregated inverter-level FFNN models *A* and *B*. . . . 91
4.16 Average NRMSE and MAPE test data set results for the macro-level PV-system and aggregated inverter-level FFNN models. 93
4.17 The 95 % Bootstrap CIs for all weather conditions for the aggregated inverter and macro-level FFNN models. 95
4.18 GRU-RNN inverter-level model development Phase-1 solution space. 96
4.19 Comparison of NRMSE and MAPE validation-set results for aggregated inverter-level GRU-RNN models C and D. 97
4.20 Comparison of NRMSE and MAPE validation-set results for aggregated inverter-level GRU-RNN models E and F. 98
4.21 Test set NRMSE and MAPE results for the macro-level and aggregated inverter-level GRU-RNN models. 100
4.22 The 95 % Bootstrap CIs for all weather conditions for the aggregated inverter and macro-level GRU-RNN models. 101
4.23 Summary of aggregated inverter-level and macro-level PV-system NRMSE and MAPE results 102
4.24 Day-time-only FFNN and GRU-RNN model test set MAPE results. 106
4.25 Day-time-only 95 % Bootstrap CIs obtained for all weather conditions for the aggregated inverter-level and macro-level PV-system FFNN and GRU-RNN models. 108

5.1 PV-system macro-level forecast results summary as determined for the test set. . 110
5.2 PV-system macro-level vs. inverter-level test-set forecast results summary. . . . 111
5.3 Final day-time-only hourly forecasting MAPE [%] results obtained. 112

A.1 Weather sensor data recorded. 116
A.2 Inverter sensor data recorded. 117
A.3 Available data points recorded for HV and MV transformers. 117
A.4 Sensor data captured by power transducer. 118
A.5 Error correlation of number of missing data points and values linearly interpolated. 122
A.6 Largest number of consecutive missing data points for the power data sets. . . . 134
A.7 Suitable inverter replacements for inverters with prolonged faulty data entries. . 136
A.8 Inverter replacements used for corresponding missing inverter data. 141

C.1 Best FFNN model architecture results as combined for various MB sizes. 150
C.2 Best LSTM-RNN model architecture results as combined for various MB sizes. . 151
C.3 Best GRU-RNN model architecture results as combined for various MB sizes. . . 151

D.1 Sliding-window size results for the FFNN aggregated inverter-level model with historic input features: P, GHI, T 154
D.2 Sliding-window size results for the FFNN aggregated inverter-level model with historic input features: P, GHI, T, WS, WD 155
D.3 FFNN *A* inverter-level model hyperparameter selections. 155
D.4 FFNN *B* inverter-level model hyperparameter selections. 156
D.5 Sliding-window size results for the GRU-RNN aggregated inverter-level model with historic input features: P, GHI, T 157
D.6 Sliding-window size results for the GRU-RNN aggregated inverter-level model with historic input features: P, GHI, T, AP 157

D.7 Sliding-window size results for the GRU-RNN aggregated inverter-level model with historic input features: P, GHI, T, WS, WD 157
D.8 Sliding-window size results for the GRU-RNN aggregated inverter-level model with historic input features: P, GHI, T, WS, WD, AP 158
D.9 GRU-RNN inverter-level model *C* Phase-1 best HL and HU combinations. . . . 158
D.10 GRU-RNN inverter-level model *D* Phase-1 best HL and HU combinations. . . . 159
D.11 GRU-RNN inverter-level model *C* Phase-2 best HL and HU combinations. . . . 159
D.12 GRU-RNN inverter-level model *D* Phase-2 best HL and HU combinations. . . . 160
D.13 GRU-RNN inverter-level model *C* hyperparameter selections. 160
D.14 GRU-RNN inverter-level model *D* hyperparameter selections. 161
D.15 GRU-RNN inverter-level model *E* Phase-1 best HL and HU combinations. . . . 162
D.16 GRU-RNN inverter-level model *F* Phase-1 best HL and HU combinations. . . . 162
D.17 GRU-RNN inverter-level model *E* hyperparameter selections. 163
D.18 GRU-RNN inverter-level model *F* hyperparameter selections. 163

E.1 Day-time-only 95 % Bootstrap CIs obtained for clear weather conditions for the aggregated inverter-level and macro-level PV-system FFNN and GRU-RNN models. 166
E.2 Day-time-only 95 % Bootstrap CIs obtained for clear-intermittent weather conditions for the aggregated inverter-level and macro-level PV-system FFNN and GRU-RNN models. 166
E.3 Day-time-only 95 % Bootstrap CIs obtained for intermittent weather conditions for the aggregated inverter-level and macro-level PV-system FFNN and GRU-RNN models. 167
E.4 Day-time-only 95 % Bootstrap CIs obtained for overcast weather conditions for the aggregated inverter-level and macro-level PV-system FFNN and GRU-RNN models. 167

F.1 Total training time required for macro-level and inverter-level models. 168

Nomenclature

Symbols and Variables

$a_i^{(L)}$	Activation output of the artificial neuron
$b_i^{(L)}$	Bias term of artificial neural network layer L and artificial neuron i
$\bar{B}$	Bootstrap estimate
B_n	Bootstrap sample
$g^{(L)}(\cdot)$	Activation function assigned to layer L
$J(\cdot)$	Cost function
P	Measured power output
$\hat{P}$	Predicted power output
$P_{capacity}$	Rated capacity of power source
P_{Inv-i}	Power output of inverter i
P_{loss}	Power losses
P_{macro}	Macro-level power output
P_{max}	maximum measured power output value
P_{norm}	Normalising power output value
$tanh$	Hyperbolic tangent activation function
$w_{k,i}$	Synaptic weight assigned to the output of neuron k delivered to neuron i
x_k	The input values received from neuron k in the previous layer.
$z_i^{(L)}$	Input summation of artificial neuron i for layer L
α	Learning rate
β	Altitude angle of sun relative to horison
σ	Sigmoid activation function
Φ_S	Azimuth angle of sun relative to North
$\frac{\partial x}{\partial y}$	Partial derivative of x relative to y
$\boldsymbol{C_t}$	New cell state vector
$\tilde{\boldsymbol{C}}_{\boldsymbol{t}}$	Candidate vector
$\boldsymbol{f_t}$	forget gate vector
$\boldsymbol{h_t}$	Recurrent neural network cell output vector for time-step t
$\tilde{\boldsymbol{h}}_{\boldsymbol{t}}$	New hidden state vector

$\boldsymbol{i_t}$	input gate vector
$\boldsymbol{o_t}$	output gate vector
$\boldsymbol{r_t}$	reset gate vector
$\boldsymbol{u_t}$	update gate vector
$\boldsymbol{W}$	Parameter matrix of weights assigned

Abbreviations

ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
AP	Air Pressure
AR	Auto-Regressive
ARIMA	Auto-Regressive Integrated Moving Average
ARMA	Auto-Regressive Moving Average
ARMAX	Auto-Regressive Moving Average eXogenous
ARX	Auto-Regressive eXogenous
BPTT	Back-Propagation Through Time
BCRF	Bias-Compensation Random Forest
CMV	Cloud Motion Vectors
CSP	Cloud Speed Persistence
DTW	Dynamic Time Warping
ED	Euclidean Distance
ERNN	Elmann Recurrent Neural Network
FFNN	Feedforward Neural Network
GA	Genetic Algorithm
GBRT	Gradient Boosted Regression Trees
GD	Gradient Descent
GHI	Global Horizontal Irradiance
GPU	Graphics Processing Unit
GRNN	General Regression Neural Network
GRU	Gated Recurrent Unit
HISIMI	Historic Similarity
HL	Hidden Layer
HU	Hidden Unit
IPL	Inverter Power Loss
k-NN	k-Nearest Neighbours
LS-SVM	Least Squares Support Vector Machine
LSTM	Long Short-Term Memory

MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percent Error
MB	Mini Batch
MBE	Mean Bias Error
MLFNN	Multi-Layer Feed Forward Neural Network
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
NRMSE	Normalized Root Mean Square Error
NWP	Numerical Weather Predictions
P	Power
PV	Photovoltaic
RBF	Radial Basis Function
RBFNN	Radial Basis Function Neural Network
ReLU	Rectified Linear Unit
RH	Relative Humidity
RF	Random Forest
RNN	Recurrent Neural Network
RT	Regression Tree
RMSE	Root Mean Square Error
SAM	System Advisor Model
SARIMA	Seasonal Auto-Regressive Integrated Moving Average
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
SVR	Support Vector Regression
T	Temperature
TSI	Total Sky Imager/Imaging
WD	Wind Direction
WS	Wind Speed
WT	Wavelet Transform
XGB	Extreme Gradient Boosting

Chapter 1

Introduction

1.1 The need for solar power forecasting

In sharp contrast to fossil fuel driven synchronous machines, energy supplied from solar Photovoltaic (PV) systems has an inherent uncertainty regarding power output. Of course, this characteristic variability originates from a high dependence on atmospheric and environmental variables. Fortunately, the uncertainty of PV power production has not prevented the commercial success of PV-system installations. This is evident from the recent REN21 Global Status Report [1], which has reported on a global PV-power capacity of 627 GW, with the addition of 115 GW during 2019 alone. However, as the total integration of more PV systems and other renewable energy (RE) power sources continues to increase, so does the volatility in terms of power supply. This unpredictability can be strenuous on the spinning reserves of electrical networks and complicates grid balancing and planning, making grid stability an issue [2]. It is this aspect of power output inconsistency, which biases power utilities against a dominant grid integration of wind and PV-based energy sources [3–5].

The boundaries of manageable uncertainty are even more constrained for advanced electrical grids, where a power purchase agreement is undertaken between the RE power producers and the public electricity providers [6]. This agreement is established based on a bidding platform, such as the European Energy Exchange [7], where it is possible to trade and bid on energy units up to 45 minutes before distribution. As defined by the purchase agreement, the RE suppliers must commit to a predetermined energy forecast [8]. With a reasonable degree of tolerance granted, RE suppliers face a financial penalty should this tolerance be undermined. Purposefully under-predicting energy production as a strategy to mitigate these penalties is naturally penalising in itself, since this results in lost profit due to unsold energy units. Given the highly ambitious and competitive environment of commercial RE power production, the long-term success of a PV system often depends on a few percentage points.

Fortunately, the negative effects of PV system volatility can be reduced to a large extent with the use of power forecasting models, which predict the expected power to be delivered within a given time horison [9–11]. With enough lead time, accurate information regarding solar energy production enables operators to plan ahead and counter the effects of variable PV power generation [11]. Forecasting further enables energy supply and demand to be effectively matched and with scheduling, spinning reserves can be integrated harmoniously in the event of a predicted shortage of PV supplied power. Essentially, forecasting aims to provide a reliable foundation for the solar-energy industry, where grid stability is maintained, RE suppliers can successfully execute their energy unit commitments and the effective distribution of RE can commence.

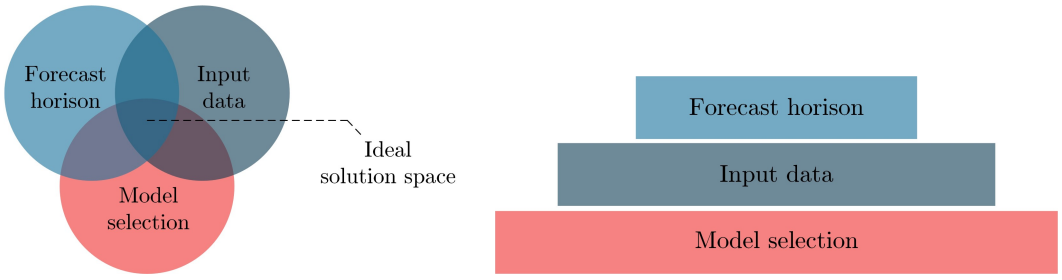
1.2 Current state of short-term PV forecasting

Before the research objectives are stated, it is necessary to have a good fundamental understanding of PV power forecasting. This is important, since the fundamentals are what categorise published works into subfields of forecasting research. With the fundamentals established, the current state of short-term intra-day forecasting as research field is presented, with an emphasis on more recently published (2015 - 2020) PV-forecasting solutions. Also, considering that the research focus is aimed at utility-scale forecasting, an overview of forecasting research performed on large ($\geq$ 20 MW) PV systems is also given. This is followed by a discussion of the most important themes, critique and knowledge gaps related to PV power forecasting. Finally, with the current state of the research field clearly outlined, the research objectives of this dissertation will be stated.

1.2.1 Forecasting fundamentals

Accurate power forecasting has become a functional prerequisite for the commercial PV sector, since it serves as a major stepping stone towards further adoption of utility-scale grid-integrated PV systems. In recent years, PV forecasting has truly emerged as a topic of great value and high priority for the commercial PV sector and academic institutions. Proof of the interest in this very active research field is evident from the extensive literature reviews published by A. Mellit and S. Kalogirou [12], J. Antonanzas *et al.* [13], M. Raza *et al.* [14], F. Barbieri *et al.* [2], U. Das *et al.* [15], S. Sobri *et al.* [16], D. van der Meer *et al.* [17] and most recently A. Mellit *et al.* [18] and R. Ahmed *et al.* [19]. This high research output is attributed to the rapid progress seen in the research field of Machine Learning (ML), which has served as a catalyst for PV forecasting as an applied research field.

PV power forecasting is also a research topic that can be defined within a multi-dimensional framework, which is another primary reason for the extensive number of research publications. To objectively examine the progress of published research, it is necessary to understand the framework, which dictates the process of forecast model development. This framework of model development is defined by 1) the forecast horison, 2) the input data and 3) the model type selection, as presented in Figure 1.1. Important to note is that there exists a natural symbiosis between these three elements as demonstrated with Figure 1.1a, which guides the process of obtaining good forecasting solutions.



(a) Model development dependencies. (b) Model development decision hierarchy.

Figure 1.1: Relationship between factors that define the framework of PV-forecasting model development.

In addition to the inter-dependency of these three factors, forecast model development also functions within a design decision hierarchy, as demonstrated by Figure 1.1b. For example, forecasts to be delivered for the desired horison of 48 h ahead, will require a certain set of

available input data (weather forecast data, historic power output, etc.). However, should the necessary input variables not be attainable, then either the forecast horison is to be adjusted or the accuracy of the obtained forecasting solution will most likely be negatively affected. It is this inter-dependence and decision hierarchy that defines the framework of probable solutions.

1.2.2 Forecast horison

As implied, the forecast horison refers to the look-ahead time, for which predictions are to be delivered. Before the current state of PV forecasting is presented, it is important to understand that the forecast horison divides this research field into several subsets of forecasting research. In accordance with the work of V. Kostylev and A. Pavlovski [20] and several other researchers [2, 21],[22, p.172], forecasting horisons are classified as presented in Table 1.1:

Table 1.1: Forecasting horison definitions.

Classification	Description	Forecast Horizon
Very short term	Now-casting Intra-hour	0 min - 30 min 30 min - 1 h
Short term	Intra-day Day-ahead	1 h - 6 h 6 h- 24 h
Medium Term	Intra-week Medium-term	24 h - 7 days 1 week - 2 months
Long Term	Long term	$\geq$ 3 months

The forecast horison is at the top of the forecast-solution decision hierarchy (Figure 1.1b). Reason being that, a forecast horison is selected based on the desired decision-making capability of those in control of and affected by the PV system [23]. For instance, intra-hour forecasts are required for rapid response scenarios, such as scheduling and deploying the spinning reserves of a network, power smoothing and dispatching [24]. Day-ahead forecasts have more value for electricity distributors concerning advanced power plant operational planning, management, load balancing, energy purchasing agreements and reducing ancillary costs. Intra-week forecasts aid with maintenance and plant management [24, 25].

Intra-day forecasts are essential for a higher resolution of generation control, real time energy unit distribution, grid balancing and energy market trading [8, 14, 26, 27]. For these reasons intra-day forecasting is the focus of this research. The distinction between day-ahead (24 h) and intra-day ($\leq$ 6 h) forecasts is due to the two energy-trading markets that exist for these intervals [17]. Readers are also referred to the work of A. Sánchez de la Nieta *et al.* [8] for an overview concerning the dynamics of the energy-trading market and the bidding processes, which gives further context to the importance of intra-day forecasting.

1.2.3 Model input data sources

With the forecast horison as primary factor for PV-forecasting model development, the feasibility of delivering forecasts for the desired horison both defines and in turn is dependent on the available input data, as demonstrated with Figure 1.1. Although, there are basic input data features appropriate for certain forecast horisons, proven to assist with improved accuracy, scholars have used various input data set combinations.

An important aspect regarding input data as a fundamental building block, which further subdivides published research solutions into different categories, is the temporal nature of the input data, defined as either historic or forecast (future) input data.

1.2.3.1 Historic input data

Historic input data strictly refers to any data source, which provides real-time and historically captured data. Similarly to how the forecast horison is chosen according to the intended goal (energy dispatch, maintenance planning, etc.) of the PV forecasts, so is the input data relevant to a particular forecast horison and PV-forecast model. Therefore, concerning the input data, forecast models are further classified as either univariate (single input variable) or multivariate (multiple input variables) [13, 28]. Literature demonstrates univariate models as adequate for forecast horisons of up to 1 h and in some cases 2 h ahead [28–30].

Forecasts made beyond 1 h, typically benefit from a multivariate input data set, which can be a collective of irradiance [W/m^2], ambient and module temperature [°C], wind speed [m/s] and etc. Notably, only highly correlated data should be considered for anticipating PV-system output, with researchers unanimous in their observations that an increase in input variables, leads to an increase in forecasting model complexity and computational expense [15, 19].

1.2.3.2 Forecast input data

To predict power output, one effectively has to predict the environment. With weather related variables typically used to anticipate power output, there are available data sources, which describe the future state of the atmosphere and environment. In Table 1.2 some of the popular data sources are provided in terms of the target forecast horison, temporal and spatial resolutions, as has been proven to be effective from literature [17, 31, 32]. A basic understanding of these different future weather data sources is a functional prerequisite to help with the formulation of a PV-forecasting solution [33]. The general consensus among researchers is that PV forecasts typically beyond a 30 min to 1 h lead-time, do indicate an improvement in forecasting accuracy as weather forecasts become more beneficial for longer forecast horisons [31].

Although important with regards to an overall understanding of PV forecasting, the work presented by this dissertation will not utilise any direct weather related forecast services or models (as elaborated on further in Section 1.7). These predictions are typically not equally available for all locations and might introduce an external element of variability.

1.2.4 Performance metrics

Before the results of published literature are presented, readers must understand and have the ability to interpret the performance metrics used to convey forecasting accuracy.

Regarding the performance metrics commonly used, past research has somewhat suffered from a lack of consistency, which has made it difficult in some cases to effectively compare model accuracies [33]. However, the issue of evaluating deterministic forecasts, also referred to as single-valued or point predictions, has to a large extent been addressed by the research community. This has lead to the wide adoption of two primary performance metrics, which are the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) [2, 13, 32]. Before the literature review continues, it is important to understand how these deterministic performance metrics are used to evaluate the time-series point predictions of published mod-

Table 1.2: Spatial and temporal correlation of the available data sources that describe the future state of the environmental and atmospheric conditions [17, 31, 32].

Data Source	Description	Forecast Horison	Temporal Resolution	Spatial Resolution
Persistence	Most basic forecasting approach, assumes no change in current weather conditions.	1 - 30 min	1 min	on-site
Total Sky Imagers (TSI)	Ground based sky imaging system used for actively monitoring sky conditions.	1 - 30 min	1 min	0 - 1 km
Satellite based	Satellite imagery provides a view from above of atmospheric conditions.	15 min - 4 h	15 - 30 min	0 - 100 km
Statistical models	Use of statistical and machine learning based techniques for data-driven weather forecasts.	15 min - 24 h	1 min - 6 h	0 - 1 km
Numeric Weather Prediction models	NWP models provide a physical description of the future state of the atmosphere.	1 h - 15 days	10 min - 10 days	1 - 100 km

els. The variables $\hat{P}_i$ and P_i in Eqs. (1.1) - (1.4), represent the predicted and the measured power values, for the i^{th} element of each data set, which both contain N elements.

The RMSE, as defined by Eq. (1.1):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{P}_i - P_i)^2} \tag{1.1}$$

This is a more aggressive metric than the MAE, due to the squared order of the error. As the forecasting error increases, so does the weight of the error penalty provided by the RMSE metric [32]. This more aggressive penalty is ideal for evaluating the overall forecast performance of a model, since large forecast errors have more severe consequences (added stress on grid and financial loss).

The Normalised Root Mean Square Error (NRMSE) is defined with Eq. (1.2):

$$NRMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\hat{P}_i - P_i}{P_{norm}}\right)^2} \cdot 100\ \% \tag{1.2}$$

This metric is useful, since normalising the RMSE metric gives researchers the ability to do a comparative analysis for different cases of PV-system installations [9]. From Eq. (1.2), the $NRMSE$ represents the $RMSE$ value, divided by the normalising term P_{norm}.

The MAE is less sensitive to outlier values in comparison to the RMSE [22, p.41][17] and is defined with Eq. (1.3) as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{P}_i - P_i| \tag{1.3}$$

From the perspective of the PV-system and electrical-grid operators, the MAE serves as a direct and easy to interpret evaluation metric. For example, a MAE of 750 kW, directly interprets as a 750 kW difference in predicted and measured power output.

The Mean Absolute Percentage Error (MAPE), as defined by Eq. (1.4), gives further perspective by providing the forecast error as a percentage value, which can be used for the

comparison of results delivered for different PV-system sizes and scenarios (such as different day types, or forecast horisons) [32].

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{P}_i - P_i}{P_{norm}}\right| \cdot 100\% \tag{1.4}$$

As mentioned, the normalisation of these error metrics, which become the NRMSE (sometimes referred to as relative RMSE) and MAPE metrics, allows researchers to evaluate and compare published forecasting studies for different PV-system sizes, relative to one another. However, it is unfortunate to see that there is still not a consensus regarding the denominator to be used for the normalisation of these error metrics. Evident from literature, researchers normalise either with the target value (P_i), the maximum installed capacity ($P_{capacity}$), the maximum measured value for the given time period (P_{max}), or the average of the measured values ($\bar{P}_i$). Some of these denominators are also mentioned by the work of T. Hoff *et al.* [34], who assessed the different approaches towards obtaining normalised metrics for irradiance forecasting accuracy.

However, with this dissertation a formal argument is made for the use of rated PV-system capacity ($P_{capacity}$) as the normalising term. By normalising with respect to system capacity, context is maintained of a real-world forecasting scenario, where large errors have more severe consequences regarding system profitability and additional stress on the electrical grid.

As an example, suppose that the NRMSE and MAPE is determined with the target value P_i as denominator. Then forecast values of 1 kW and 5 MW, for respective target values (P_i) of 2 kW and 10 MW, will translate to a 50 % MAPE in both cases. However, for a utility-scale scenario a 5 MW error, compared to a 1 kW error, will have a much larger impact on the electrical grid and it is important that metrics convey this when model accuracies are evaluated. Therefore, with the NRMSE and MAPE determined relative to PV-system capacity, perspective is maintained.

1.2.5 PV-forecasting model selection

Once the forecast horison and preferred input data sources have been established, the forecast model type is to be selected.

1.2.5.1 Deterministic and probabilistic models

PV-forecasting models are firstly categorised as either deterministic, where point predictions are delivered, or as probabilistic, where an interval of values are provided together with a quantified uncertainty. Given that the majority of PV-forecasting literature has and continues to be predominantly focused on delivering deterministic point predictions, the focus of this dissertation remains on deterministic models.

However, some authors [17, 35, 36] have argued for the use of probabilistic PV-forecasting models, or at the very least, to extend the evaluation of deterministic models to contain probabilistic information. Reason being that a quantified uncertainty further aids with the framework for decisions and improved risk-management strategies to be executed [17, 37]. An example of extracting probabilistic information from deterministic forecasts is found in the work of H. Wang *et al.* [35], who implemented Quantile Regression (QR) to evaluate the probabilistic information in the PV power data. The use of lower upper bound estimates (LUBE), originally proposed by A. Khosravi *et al.* [38] has also been widely adopted to establish prediction intervals, as seen in the wind, load and PV power forecasting fields

[17, 36, 39]. Another very familiar method for generating probabilistic forecasts from deterministic models is with an ensemble approach. The work of S. Sperati *et al.* [40] for example used an ensemble of Artificial Neural Networks (ANNs) to deliver probabilistic PV power forecasts. Authors Q. Ni *et al.* [36, 41] further delivered probabilistic forecasts with a unique combination of both the mentioned LUBE and ensemble methods.

Although the argument for quantifying uncertainty by means of probabilistic models is valid, solutions such as an ensemble of traditional neural networks seen in the work of A. Khosravi *et al.* [42] and S. Sperati *et al.* [40] for example, have a high computational cost, which may be a problem for delivering real-time forecasts [35, 43]. Researchers have attempted to reduce the computational cost of such ensemble models, as for example presented by the work of Q. Ni *et al.* [41] who proposed the ensemble of less computationally expensive extreme learning machine models. However, the reduced computational expense can primarily be attributed to the fact that this was essentially a shallow ML-based solution.

The reason for maintaining the focus of this dissertation on deterministic models is due to the ability of stand-alone deep learning (DL) models to extract complex non-linear relationships between model input and output variables [35, 44]. Also, most of the innovations in the field of PV forecasting and ML-based time-series forecasting are focused on deterministic models. Due to this focus on point predictions, there is an expectation that future research will likely be more focused on the extraction of uncertainty information from deterministic forecasts. With examples seen of how DL models can evidently be further expanded to deliver probabilistic forecasts, as seen in the two unique approaches of H. Wang *et al.* [45] and H. Wang *et al.* [35]. Also, popular methods such as the LUBE approach, have proven to be a valuable complimentary source of deriving uncertainty information from point forecasts [46]. For these reasons, the research focus is maintained on the individual assessment regarding the ability of deterministic, stand-alone DL models to extract low-level PV-system behaviour.

1.2.5.2 Forecast model classification

With models either developed to deliver point or interval forecasts, another primary distinction is the classification of models into three main approaches, as illustrated in Figure 1.2.

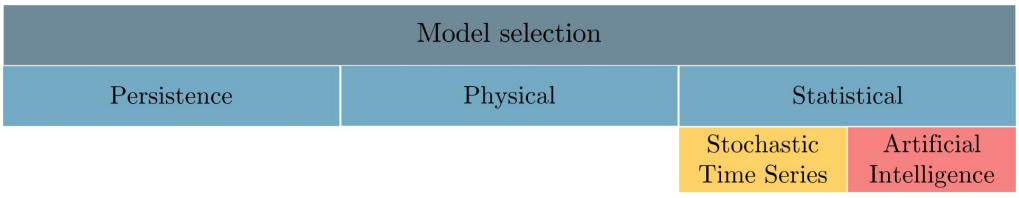


Figure 1.2: Classification of the various PV forecasting techniques.

As previously defined, the research presented with this dissertation is focused on intra-day (1 - 6 h) forecasting with forecasts delivered only using historic input data. Given this research framework and the classification of forecast model techniques, the majority of published intra-day forecasting research has used statistical models.

Statistical modelling refers to forecasting methods, which rely on historical time series and real-time data, so that a data-driven prediction anticipates the system behaviour (response variable) [13, 19]. Thus, these models establish the relationship between PV-system power output and a set of input variables. As guided by Figure 1.2, statistical models can be further categorised as stochastic time-series and artificial intelligence (AI) models.

In terms of stochastic time-series models, the autoregressive models have proven to be a prevalent choice, such as the Auto Regression (AR) and the Auto Regression Moving Average (ARMA) models, along with variants of these models, such as ARX and ARMAX, where 'X' represents the inclusion of exogenous data entries [13]. For linear, non-stationary data sets, literature indicates the most popular model to be the Auto-Regressive Integrated Moving Average (ARIMA) model (also known as Box-Jenkins model), which is essentially the summation (integration) of an ARMA process [47]. The ARIMA model is a popular model, because unlike the ARMA model, the ARIMA model does not require the time-series data set to be stationary.

A popular extension of the ARIMA approach is called SARIMA, with the 'S' denoting Seasonal-ARIMA. This allows the ARIMA method to be applied to a time-series data set, where data is classified according to seasonal variations during the year. Ultimately, the AR, ARMA and ARIMA models are better suited for short-term forecasts ranging between 15 min - 1 h ahead, where less input parameters (meteorological) are available [14, 31]. Although useful, these models and other variants of these models have been consistently outperformed by ML-based models where forecasts beyond 1 h ahead were the focus [9, 48–50]. Thus, with the forecast horison of 1 - 6 h ahead for this dissertation, these models are not to be considered for this research.

The overwhelming majority of recent literature has focused on and has proven the efficacy of AI-based models to deliver forecasts for exactly this forecast horison, truly making AI models the state-of-the-art for this research segment of PV power forecasting.

1.2.6 Artificial intelligence techniques

Artificial Intelligence (AI) models have proven to be the state-of-the-art in the field of PV forecasting. These AI models are further classified as ML and DL models, which are a subset of AI techniques, as given context with Figure 1.3.

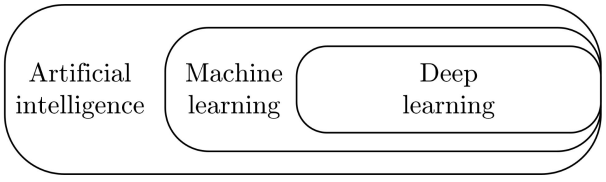


Figure 1.3: Classification of artificial intelligence techniques.

The focus of this section is not to provide a theoretical foundation for each of these models, but is instead dedicated to mention the most prominent models from literature and how these models compare. More theoretical depth will be provided accordingly in the chapters to follow, since most of the model design choices are influenced by the theory.

Finally, given the multi-dimensional framework that classifies PV forecasting research into different categories, only published forecasting results are considered, which fall within the same research framework as the work of this dissertation. Specifically, these are studies that utilised historic input data (no weather forecast data) and delivered intra-day forecasts with a temporal range anywhere between 1 - 6 h ahead.

1.2.6.1 Machine learning models

The ability of ML models to extract and emulate the non-linear relationships between input features and PV-system power output has resulted in a plethora of ML-based research

solutions. Historically, literature preceding the year 2017 indicates that Support Vector Machines (SVMs) and shallow (one hidden layer) Feedforward Neural Networks (FFNNs), also referred to as Multi-Layer-Perceptrons, were frequently adopted as the most prominent ML approaches in the field of both PV power [13–15] and irradiance forecasting [25, 26, 31, 51].

There are many examples of research that applied SVMs, also referred to as Support Vector Regression (SVR) models, within the same research framework of this dissertation [28, 52, 53]. For example, the research by M. Rana *et al.* [28] presented a SVM model as benchmark, compared to an ensemble of shallow FFNNs as the proposed model. Both models were compared in a multivariate and univariate analysis, which in this case proved the FFNN model to be more accurate. The research by K. Li *et al.* [52] delivered forecasts 30 min - 6 h ahead with an Extreme Gradient Boosting (XGB), FFNN, Random Forests (RF) and SVM model. In terms of the single-valued forecasts, the proposed XGB model proved to be superior. Interestingly, although not more accurate than the XGB model, the SVR model did outperform the RF and FFNN models.

Another extremely popular variant of the SVM model, has been the use of a Least Squares SVM (LS-SVM) model first introduced by J. Suykens and J. Vandewalle [54]. Essentially, this SVM model simply utilises a quadratic cost function, which results in a computationally less expensive model, ideal for large data set problems. This improvement in the SVM performance lead to the wide adoption of the LS-SVM, as seen in the works of M. De Giorgio *et al.* [55] and Y. Wu *et al.* [9] for example.

The use of Wavelet Decomposition, sometimes interchangeably referred to as Wavelet Transform (WT), in combination with classic ML-based models has also proven to be a popular technique to further improve model performance. The process of Wavelet Decomposition involves decomposing the historic power output signal into time-frequency representations. Therefore, the historic power output signal is typically divided into low and high time-series frequency components, represented as continuous time signals, which are then delivered as input features for the model. Another typical approach has also been to train individual models on each of these decomposed power output signals and then to reconstruct these outputs into a single forecast time-series signal. Examples of this approach are found in the works of M. De Giorgio *et al.* [55] and H. Wang *et al.* [35]. The research by D. AlHakeem *et al.* [56] used WT as a pre-processing step to manage the hourly time-series data fluctuations. The models compared were a WT+FFNN, WT and Generalized Regression Neural Network (WT+GRNN). With the data provided to the models, the forecasts delivered for each frequency component were also reconstructed to deliver the final forecast.

Apart from the fact that SVMs and FFNNs have dominated the PV forecasting arena, other ML approaches were also frequently adopted, such as the use of Regression Trees (RTs), and RF models [52, 57–59]. For example, a unique approach was the novel Bias-Compensation Random Forest (BCRF) model, with a WT applied to the input data, as proposed by P. Chiang *et al.* [57]. The authors attributed the success of the model to the various frequency components obtained from the WT, which allowed the BCRF model to order the influence of the wavelet components, given the different forecast horisons. Likewise to K. Li *et al.* [52], the research by C. Persson *et al.* [58] also presented a gradient-boosted approach. They applied a Gradient Boosted Regression Trees (GBRT) model for the multi-site prediction of future power generation from 42 PV rooftop installations. ML-models such as Elman-Recurrent Neural Networks (Elman-RNNs) [55, 60] and neuro-fuzzy networks [9] were also frequently adopted to emulate the non-linear relationships between power output and input data within the PV forecasting arena.

Also determined to compare a SVM-based model to a shallow FFNN model, Z. Li *et al.*

[61] made predictions for a 6 MW rated PV power plant, located at Florida, USA. Interestingly, the authors did not make use of irradiance as input, but instead utilised information regarding the position of the sun, together with ambient temperature, wind direction and speed. With the 6 MW rated PV system consisting of 11 inverters, the authors trained models for each inverter. From the aggregated inverter-level forecasts delivered 1 h ahead by the shallow FFNN and SVR models, a marginal improvement was recorded as compared to the macro-level models. Ultimately, the NRMSE accuracies were determined as 13.1 % and 13.2 % for the FFNN and SVR models, respectively.

A combination of different forecasting models is usually superior due to an ability to overcome the characteristic limitations of one single technique [25, 62]. This is the reason why Hybrid forecasting models are developed, which seek to combine the advantages of two or more different forecasting methods. From published research there are many acclaimed hybridised short-term PV-forecasting models, all of which attempt to improve on the results of a single forecasting technique [4, 9, 30, 48, 55, 56, 63, 64].

From literature, it is further noted that researchers often wrongfully declare the use of hybrid solutions, as seen with some of these publications. For example, several studies have claimed the use of hybrid models, where a WT technique has been used to decompose the historic power output signal into different time-series frequency components, as seen in the works for example by P. Mandal *et al.* [4], H. Zhu *et al.* [63], M. De Giorgio [55] and several others. This classification of models as being hybrid, due to a use of the WT is incorrect, since the WT serves as a data pre-processing technique and is not capable of delivering forecasts as a stand-alone model. Also, the use of a Genetic Algorithm (GA) or Particle Swarm Optimisation (PSO) as optimisation techniques is also not strictly speaking a hybrid solution, since these techniques cannot function as stand-alone forecasting models.

An example of where the use of a GA has served to deliver a hybrid approach can be found in the work of Y. Wu *et al.* [9]. They provided 1 h ahead forecasts for three different PV systems, using a GA as a final weighted adjustment for the forecasts received from an ARIMA, LS-SVM, ANN and ANFIS model. The research by M. Bouzerdoum *et al.* [64] also serves as a good example of a true hybrid approach. They further enhanced their forecast accuracy with the application of a SARIMA-SVM hybrid model to forecast PV power 1 h ahead. The SARIMA model was used to estimate the linear components of the power output data, whereas the SVM model served to establish the non-linear patterns of the SARIMA calculated residual series. Other prominent examples of pure hybrid models can be found in the mentioned work of K. Wang *et al.* [44] and W. Lee *et al.* [65], where both studies were executed with a hybrid Convolutional Neural Network and Long Short-Term Memory (CNN-LSTM) model.

Although these aforementioned ML-models are no longer considered to be state-of-the-art, these models continue to be applied in PV-forecasting literature and are to a great extent still relevant. Therefore, some of the most noteworthy ML-based PV forecasting publications are summarised in Table 1.3.

Table 1.3: Summary of the short-term forecasting literature results obtained with machine learning models as solution.

Authors	Location	Rated Capacity	Forecast Horizon	Accuracy	Normaliser P_{norm}	Forecast Model	Description
P. Mandal (2012) [4]	Oregon, USA	15 kWp	**1 h:** 1 h	**MAPE:** Clear: 2.38 % - 6.31 % Cloudy: 4.08 % - 20.47 % Rainy: 4.5 % - 16.71 %	$\bar{P}_{target}$	WT + FFNN	Hybrid model of WT + RBFNN superior to FFNN, RBFNN and WT+FFNN models. Used historical and actual irradiance, temp. and power data. Results presented also account for seasonal variations.
M. Bouzerdoum *et al.* (2013) [64]	Trieste, Italy	20 kWp	**1 h:** 1 h 1 h	**NRMSE:** 9.4 % **MPE:** 2.74 %	P_{max} - P_{min}	SARIMA + SVM	The SARIMA+SVM hybrid model was proved superior to individual SARIMA and SVM models. Used historic irrad., power, ambient and module temp.
C. Monteiro *et al.* (2013) [66]	Spain	2.8 MWp	**1 h - 24 h:** 1 h	**NRMSE:** 10.14 %	P_{max} - P_{min}	HISIMI + GA	Proposed a persistence-based HISIMI+GA model. The model proved superior compared to FFNN and persistence models. Used historic power output and NWP supplied irradiance temperature as input data.
Y. Wu *et al.* (2014) [9]	Taiwan, Malaysia	3x PV systems: System1: 45 kWp System2: 72 kWp System3: 70 kWp	**1 h:** 1 h	**NRMSE:** System1: 5.64 % System2: 3.43 % System3: 6.57 %	$P_{capacity}$	ARIMA+ +LS-SVM+ ANN+ ANFIS+ GA	Proposed a hybrid ARIMA+ LS-SVM +ANN+ ANFIS GA model. The GA was used to obtain a weighted aggregated output, with each model assigned a level of influence. The LS-SVM had the most influence. Used historic power, ambient and module temp. and irrad. from NWP as input data. Proposed model was superior compared to all of the stand-alone models.
M. Rana *et al.* (2016) [28]	Brisbane, Australia	4 x PV Systems: 1.22 MWp total	**5 min - 1 h:** 5 min 15 min 30 min 45 min 1 h	**MAPE:** 4.15 % 6.28 % 7.46 % 8.26 % 8.71 %	P_{target}	ANN Ensemble	Univariate (power only) and Multivariate (irradiance, temp., humidity and wind speed) input data was used. The univariate approach was proven most accurate, with ANN ensemble and SVR delivering the best results. The ANN ensemble model was proven to be most accurate.
Z. Li *et al.* (2016) [61]	Florida, USA	6 MWp	**15 min - 24 h:** 1 h 1 h	**NRMSE:** 13.4 % **MAPE:** 4.31 %	$\bar{P}_{target}$ Max(P_{target})	FFNN	Hierarchical forecasting approach, with FFNN and SVR models compared. Used historic inverter level power output, ambient temp., wind speed and direction, and solar position angles as input data. From the relative NRMSE results, marginal improvement was seen for aggregated micro-level forecasts.

Table 1.3: Summary of the short-term forecasting literature results obtained with machine learning models as solution.

Authors	Location	Rated Capacity	Forecast Horizon	Accuracy	Normaliser P_{norm}	Forecast Model	Description
H. Wang *et al.* (2017) [35]	Belgium	2x PV systems: Unknown	**15 min - 2 h:** 45 min 75 min	**MAPE:** 2.74 % - 5.77 % 2.25 % - 4.97 %	$\bar{P}_{target}$	WT + CNN + QR	A deterministic model was developed with the combination of WT and CNN techniques. Quantile Regression (QR) was used to further adapt the model for probabilistic forecasts. Input data consisted of power, historic temp., rainfall, sunshine hours, max. wind gust and average solar irradiance. The proposed model was compared to FFNN, SVM and WT+SVM models. The 45 min results range across 12 months, with 75 min results averaged for seasons.
M. Hossain *et al.* (2017) [53]	Malaysia	3x PV systems: System1: 2 kWp System2: 1.9 kWp System3: 2.7 kWp	**1 h - 24 h:** 1 h	**MAPE:** 0.35 % - 1.70 %	P_{target}	ELM	Compared an ELM model to a FFNN and SVR model. Results compared computational time and accuracy for various training and data set combinations. Used hourly averaged irradiance, module and ambient temp., and wind speed as model input data.
H. Pedro & C. Coimbra (2012) [48]	California, USA	1 MWp	**1 h - 2 h:** 1 h 2 h	**NRMSE:** 6.4 % - 16.72 % 8.99 % - 24.39 %	P_{target}	GA + FFNN	Applied an univariate forecasting approach, where models compared only used historic output power data. GA+ANN model was proven superior to ARIMA, k-NN and stand-alone FFNN models. Results provided here are for days with different levels of intermittence.
M. Rana *et al.* (2015) [67]	Brisbane, Australia	4 x PV Systems 1.22 MWp total	**30 min - 6 h:** 30 min 1 h 2 h 3 h 4 h 6 h	**MRE:** 6.47 % 8.58 % 11.05 % 11.14 % 11.40 % 9.26 %	–	SVR2D	Introduced a new probabilistic SVR2D model. Used multivariate input data set of previous power output, irradiance, temp., humidity and wind speed. The SVR2D model proved superior to the MLFNN2D model, as well as the immediate and historic day persistence models. The interval coverage probability was averaged to 73.1 %.
K. Li *et al.* (2018) [52]	Queensland, Australia	433.33 kWp	**5 min - 6 h:** 5 min 30 min 1 h 2 h 6 h	**MAPE:** 9.6 % 12.5 % 13.8 % 15.4 % 15.7 %	–	XGB	Applied a novel Envelope-based clustering technique to classify irradiance into subsets, used for model training and forecasting. Also used temperature, humidity, wind speed and direction, month, day and hour as input features. The XGB model proved to be superior as compared to RF, FFNN and SVR models. The proposed Improved Bootstrap prediction interval construction method proved superior to the traditional bootstrap, SVR-2D and LUBE estimation models.

Table 1.3: Summary of the short-term forecasting literature results obtained with machine learning models as solution.

Authors	Location	Rated Capacity	Forecast Horizon	Accuracy	Normaliser P_{norm}	Forecast Model	Description
M. De-Giorgi *et al.* (2014) [55]	Puglia, Italy	960 kWp	**1 h - 24 h:** 1 h 3 h 6 h 1 h 3 h 6 h	**NRMSE:** 9.6 % 14.09 % 15.28 % **MAPE:** 6.92 % 10.35 % 10.53 %	Max(P_{target})	WT + LS-SVR	Used a 1 h avg. of 10 min samples of ambient and module temp., irradiance and historic power The hybrid WT+LS-SVR model was more accurate than the individual FFNN and LS-SVM models.
D. AlHakeem *et al.* (2015) [56]	Oregon, USA	15 kWp	**1 h - 6 h:** 1 h: 3 h: 6 h: 1 h: 3 h: 6 h:	**NRMSE:** 1.15 %-27.87 % 3.33 %-40.91 % 6.82 %-45.71 % **MAPE:** 0.87 %-20.74 % 2.06 %-36.64 % 5.15 %-40.94 %	$P_{capacity}$	WT + GRNN + PSO	Used PV power output, irrad., and temp. as input. Applied a WT to decompose PV input data. The GRNN model was optimised with a PSO algorithm. Results here are provided for all seasons and clear, cloudy and overcast day types. The WT+GRNN +PSO model proved superior to WT+FFNN and WT+GRNN models.
C. Persson *et al.* (2017) [58]	Japan	42x systems: 2.4 - 9.6 kWp	**1 h - 6 h:** 1 h 6 h	**NRMSE:** 10 % 13.7 %	Max(P_{target})	GBRT	Utilised Gradient Boosted Regression Trees (GBRTs) as multi-site predictive model. Forecasts made for 42 rooftop installations. Results in favour of the multi-model as compared to the single site GBRT models. The persistence, climatology and recursive AR models were outperformed by the proposed model.
M. Malvoni *et al.* (2017) [68]	Italy	2x PV systems: System1: 353.3 kWp System2: 606.7 kWp	**1 h - 24 h:** 1 h - 6 h	**NRMSE:** 1 % - 6.6 %	Max(P_{target})	GLSSVM + PCA + WT	Applied a Group Least Squares SVM model. Inputs consisted of ambient temp., POA irradiance, wind speed and PV power. As pre-processing step, PCA & WT techniques were used in various combinations. The WT applied to the principal components was verified as the superior approach.
P. Chiang *et al.* (2017) [57]	California, USA	Unknown	**1 h - 6 h:** 1 h 3 h 6 h	**MAPE:** 6.33 % 6.06 % 5.91 %	P_{target}	WT + and BCRF	With solar irradiance, UV index, temperature and humidity at 2.5 min intervals as input, a WT bias compensated random forest model was developed. This model proved most accurate compared to the FFNN, WT+SVM and RF models.
A. Eseye (2017) [69]	Beijing, China	480 kWp	**3 h - 24 h:** 3 h 6 h	**MAPE:** 0.87 % - 1.78 % 1.2 % - 2.45 %	$\bar{P}_{target}$	WT + SVM + PSO	Developed a hybrid WT and SVM model, optimised by a PSO technique. Historic PV-system power, irrad., ambient , cloud cover, humidity, pressure and wind speed as input. Model proven superior to seven models: BPNN, GA+ANN, SVM, GA+SVM, SVM+PSO & Hilbert-Huang--Transform+SVM +PSO model.

Table 1.3: Summary of the short-term forecasting literature results obtained with machine learning models as solution.

Authors	**Location**	**Rated Capacity**	**Forecast Horizon**	**Accuracy**	**Normaliser** P_{norm}	**Forecast Model**	**Description**
X. Agoua *et al.* (2018) [59]	France	PV data set 1: 45 kWp - 5MWp PV data set 2: 3.2 kWp - 58 kWp	**15 min - 6 h:** 6 h	**Improved RMSE:** 6 % (over RF) 10 % (over AR)	–	Spatio-Temporal	Proposed the stationarisation of raw time-series power output data. Also presented a spatio-temporal model, which relied on data input from various geographically dispersed PV systems. With AR and RF as benchmark models, the RMSE results indicated here present the average improvement over the AR and RF models, respectively, for a 6 h forecasting horizon.

1.2.6.2 Deep learning models

Similar to ML, DL is a mathematical framework for learning representations from data, but mainly refers to the stacking of more layers, which increases representational power [51, 70]. Ultimately, the need for more powerful models has resulted in the abrupt shift of recent PV-forecasting literature towards DL models. With increasing data set sizes and more available input features, DL models are ideal for PV forecasting applications, since these models overcome the shortcomings of shallow ML models, because of an enhanced feature-value extraction ability [35, 44].

Common DL neural networks mainly include deep FFNNs, Deep Belief Networks, Stacked Auto-Encoders, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [18, 19, 44, 51]. Considering recent publications [44, 51] and the latest literature reviews completed by A. Mellit *et al.* [18] and R. Ahmed *et al.* [19], DL models to be considered as the state-of-the-art within the PV-forecasting research field are deep FFNNs (with two or more hidden layers [51, 70]), RNNs and CNNs. With RNNs ideal for processing sequences of time-series data to deliver forecasts, two RNN-based models, which have proven to be effective as DL models, are the Long Short-Term Memory RNN (LSTM-RNN) and the Gated Recurrent Unit RNN (GRU-RNN) models.

With the focus of this dissertation on state-of-the-art DL models to deliver intra-day forecasts from historic data only, this section is focused on recent (past 3 years, with exceptions) literature published within the same research framework. One of the first publications to use an LSTM-RNN as PV power forecasting model was the 2017 work of M. Abdel-Nasser and K. Mahmoud [29]. With the PV systems located in Egypt, only historic power output (univariate model) was used to deliver the 1 h ahead forecasts. From the results, the LSTM-RNN model outperformed the FFNN with two hidden layers.

The LSTM-RNN model of D. Lee *et al.* [71] was also compared to a shallow one hidden layer and a deep seven hidden layer FFNN. With data from a 20 kW rated PV system located in Gumi, South Korea, models were trained with temperature, humidity, irradiance, cloud index, month and day of the month as input features. Multi-step 14 h ahead forecasts were delivered, at a 1 h resolution. The LSTM-RNN outperformed the deep FFNN results. With several sliding window sizes ranging between 1 - 6 h applied, a total improvement of 53 % regarding the MAE was achieved by the LSTM-RNN model over that of the FFNN results.

Another multi-step forecasting solution, which delivered 1 h ahead forecasts at a 5 min resolution, was presented by the recent work of P. Li *et al.* [72]. With the use of only historic weather and power data, wavelet packet decomposition was applied to deconstruct the original power output signal into low and high frequency components. With an independent LSTM-RNN trained on each subset, the forecasts were aggregated with a final linear combination applied. The proposed model outperformed the LSTM-RNN, GRU-RNN, RNN, SVM, and FFNN models for all seasons of the year.

With utility-scale forecasting as objective, the first authors to claim the use of a Bidirectional LSTM-RNN (Bi-LSTM) was presented by H. Sharadga *et al.* [50]. With forecasts delivered 1 - 3 h ahead, an extensive comparison of the Bi-LSTM was made to a regular LSTM-RNN, FFNN models, ARMA, ARIMA and SARIMA models. Similar to many other studies, results once again proved the superiority of ML-based models, especially as forecast horisons extend beyond 1 h ahead. Compared to the LSTM-RNN and FFNN models, the Bi-LSTM forecasts were most accurate.

In an attempt to further enhance the LSTM-RNN forecasting ability, the first use of the Attention mechanism combined with an LSTM-RNN model was recently presented by H.

Zhou *et al.* [49]. The Attention mechanism [73] is a relatively recent ML advancement, which allowed the authors to adaptively focus on more influential input features. With historic power and temperature data used, multi-step intra-hour forecasts were also delivered with a shallow FFNN and a two hidden layer LSTM-RNN model. Ultimately, the Attention-LSTM model performed best, followed by the stand-alone LSTM-RNN model.

Although aimed at 24 h ahead forecasts, the first use of a GRU-RNN model was published by the 2018 study of Y. Wang *et al.* [74]. With forecasts delivered for three PV systems located in Australia, only using historic environmental data (temperature, humidity, irradiance, etc.), the GRU-RNN and LSTM-RNN results were very close and outperformed the shallow FFNN model. Since then the GRU-RNN has seen an early adoption from other researchers. For example, the recent research of N. Sodsong *et al.* [75] used a GRU-RNN model to deliver 1 h ahead forecasts for a PV system in Ishikawa, Japan. Only using historic measurements (power, irradiance, temperature, etc.) as input, they attempted to further reduce GRU-RNN model complexity and therefore training time. This was achieved by consecutively feeding the prediction output of smaller GRU-RNN models, together with the historic data, into another reduced GRU-RNN model. This process was repeated during training until the optimal number of GRU-RNN models, feeding into one another was established. This unique approach outperformed the stand-alone GRU-RNN model with more hidden layers. However, in comparison to the SVR, shallow FFNN and K-Nearest Neighbour models, the stand-alone GRU-RNN also outperformed these models.

Finally, the use of CNNs has also been successfully adopted within the PV-forecasting community. The 2017 work of H. Wang *et al.* [35] presents the first PV power forecasting application of a CNN. They proposed a WT and CNN model combination (WT-CNN) to deliver deterministic forecasts, which was further extended with QR to deliver probabilistic forecasts. Similar to the mentioned work of P. Li *et al.* [72], the WT deconstructed the historic power output into several time-series subsets of frequency components. This subset of signals then served as input for the CNN model. With results compared to shallow FFNN, SVM and WT-SVM models, the proposed WT-CNN model outperformed these models.

Although not within the scope of this literature review, due to differences in forecast horison and use of future weather data as input, CNNs have also been applied to the recent works published by K. Wang *et al.* [44] and W. Lee *et al.* [65]. The research executed in South Korea by W. Lee *et al.* [65] used the LSTM-RNN for long-term dependencies and two CNNs, each with a different filter size, for short-term feature extraction. With several sliding window sizes of historic data, ranging between 1 - 6 h, the CNN-LSTM model delivered the best forecasts. Unfortunately no comparisons were made to deep FFNN or stand-alone LSTM-RNN models, but the model did outperform another hybrid Autoencoder and LSTM-RNN model, first presented by A. Gensler *et al.* [76], which also used weather forecasts as input.

The most recent research being that of K. Wang *et al.* [44] developed a hybrid CNN-LSTM model, where the CNN was used to extract the spatial features and the LSTM-RNN to obtain the temporal dependencies. Interestingly, with a historic time series data length of 1 year, the LSTM-RNN model performed best. However, with a 3 year historic data set to train on, the CNN-LSTM model outperformed the LSTM-RNN and CNN models.

Both of these studies also mentioned the high computational expense of the CNN-LSTM models, in comparison to the stand-alone LSTM-RNN model.

Table 1.4: Summary of the short-term forecasting literature results obtained with deep learning models as solution.

Authors	Location	Size	Forecast Horizon	Accuracy	Denominator P_{norm}	Forecast Model	Description
H. Wang *et al.* (2017) [35]	Flanders & Limburg, Belgium	Unknown	**15 min - 120 min:** 45 min 75 min	**MAPE:** 3.85 % 3.82 %	$\bar{P}_{target}$	WT + CNN	Presented the first PV power forecasting application of a CNN. Delivered deterministic forecasts with a WT+CNN model and probabilistic forecasts with Quantile Regression. Proposed model outperformed shallow FFNN, WT-SVM models.
M. Abdel-Nasser and K. Mahmoud (2017) [29]	Egypt	Unknown	**1 h:** 1 h	**RMSE:** 82.15 136.87	–	LSTM	Five LSTM-RNN model variations executed on two PV-systems in Egypt. The LSTM-RNN models outperformed the multiple linear regression (MLR) bagged regression trees (BRT), and FFNN models.
H. Zhou *et al.* (2019) [49]	Shaoxing, China	20 kWp	**7.5 min - 1 h** 1 h 1 h	**MAPE:** 37.82 %	P_{target}	LSTM + Attention	LSTM model was combined with the Attention mechanism. Used module temp. and power to deliver 7.5 min to 60 min forecasts. Model outperformed ARIMAX, MLP, LSTM-RNN.
N. Sodsong *et al.* (2019) [75]	Ishikawa, Japan	Unknown	**1 h:** 1 h	**NRMSE:** 9.64 %	$P_{capacity}$	GRU	Used historic power output, GHI, and Temp. as input data. Unique GRU-RNN model proposed outperformed regular GRU-RNN, FFNN, SVR and k-Nearest Neighbour models.
D. Lee *et al.* (2019) [71]	Gumi, South Korea	20 kWp	**1 h - 14 h**	**MAE:** 53 % improvement over deep FFNN	–	LSTM	Delivered Multi-step 14 h ahead forecasts at 1 h resolution. With temp., humidity, irradiance and cloud index data as input, LSTM-RNN outperformed single & multi-layer FFNN.
P. Li *et al.* (2020) [72]	Alice Springs, Australia	26.5 kWp	**1 h:** 1 h	**MAPE:** 2.4 %	P_{target}	LSTM	Delivered Multi-step 1 h ahead forecasts at 5 min resolution. Used Wavelet decomposition to obtain subsets of power signal. With LSTM-RNN applied to each signal, the aggregated forecasts outperformed the regular LSTM-RNN, GRU-RNNs, RNN and SVM models.
H. Sharadga *et al.* (2020) [50]	Unknown	20 MWp	**1 h - 3 h:** 1 h	**RMSE:** 0.791 (normalised data set)	–	Bi-LSTM	First PV power publication to use Bidirectional LSTM-RNN. Forecasts delivered for 1 h to 3 h ahead. With historic data only, Bi-LSTM outperformed regular LSTM-RNN, FFNN, ARIMA, SARIMA models.

1.2.7 Utility-scale PV-system forecasting

Evident from published research is that the majority of studies have been executed on small-scale (< 100 kW) PV systems. Studies performed on large (≥ 20 MW) utility-scale systems are regarded as a rarity, which to some extent, has resulted in a disconnect between published research and utility-scale PV power production.

Compared to the vast collection of research performed on small-scale PV systems, only a total of five studies related to large PV-system forecasting could be found. The works presented by B. Urquhart *et al.* [77], M. Lipperheide *et al.* [78] and Y. Chu *et al.* [30] were all performed on the same PV system with a rated 48 MW capacity. To date, this is the largest stand-alone PV system used for forecasting research.

The 2013 study by B. Urquhart *et al.* [77] was the first to present sky-imagery as input obtained from two TSIs, to deliver PV forecasts 15 min ahead. Similar to B. Urquhart *et al.* the work of Y. Chu *et al.* [30] also used two TSIs, which captured images over a 16 km^2 area of cloud movement above the same 48 MW PV system, at 30 s intervals. Forecasts were delivered 5 - 15 min ahead with a FFNN model further optimised with a GA. Without the use of TSIs, M. Lipperheide *et al.* [78] proposed a Cloud Speed Persistence (CSP) model. From the gradual variation in PV module output, they were able to anticipate cloud speed, direction and the overall reduction in power output for a short time ahead. With forecasts delivered 3 min ahead at 20 s intervals, the proposed CSP model managed to outperform the regular persistence model. Evidently, all three of these studies were aimed at delivering very short-term forecasts, with the analysis specifically focused on cloudy and overcast days.

The research presented by J. Liu *et al.* [79] was aimed at delivering forecasts 24 h ahead for a large 20 MW PV system. With temperature, humidity, and wind speed as input, the novelty of their research was the additional use of an Aerosol index as input feature. With the use of a shallow FFNN model, the largest improvement in accuracy, due to the aerosol index, was seen for cloudy days.

Finally, the previously mentioned research of H. Sharadga *et al.* [50] serves as the most recent study to deliver forecasts (1 - 3 h) for another large PV system rated at 20 MW, with a DL-based solution.

There are other studies, which have claimed rated capacities in excess of 20 MW, such as the work of L. Gigoni *et al.* [80] and M. Pierro *et al.* [81]. However, these studies were focused on distributed generation, which collectively analysed output from various smaller PV systems at different locations. Therefore, these studies do not convey specific insights regarding forecasts delivered for large stand-alone PV systems.

1.2.8 Critique and summary of PV-forecasting research

Forecasting the power output of a grid-connected PV power plant has evidently become a research topic of tremendous interest. Given the plethora of published PV-forecasting research, almost all researchers claimed the superiority of their suggested models. With these models distinguished from one another based on forecast horison and input data, it has been identified that these claims of superiority are mostly founded on what is often completely different scenarios. There still exists a great disparity within the research field regarding these claims of superiority, which is attributed to various factors:

- *Difference in performance metrics utilised*: The use of specific evaluation metrics and a failure to substantiate these metrics, which are often favourable towards the proposed model results, further resulting in confirmation bias.

- *Benchmarking*: The comparison of proposed models against less capable models such as deep vs. shallow ML-based models.
- *Test data set length*: Final model results are presented based on a reduced test data set length (less than 1 year). Thus, results do not account for all weather types and seasons, each with a unique degree of variability. This leads to sample bias and undermines the generality of the proposed model solutions.
- *Model optimisation*: Unequal effort applied to the training and optimisation of benchmark or competitive models.

The same sentiment concerning claimed model superiority is also shared by G. Yagli *et al.* [82], M. Husein *et al.* [83] and S. Sala *et al.* [84]. However, given this disparity in acclaimed superior model performances, there are still some important movements within the research field to take note of. The current state of intra-day PV forecasting can further be summarised as follows:

1. PV-forecasting research published from 2017 and earlier has proven ANN-based and SVM-based forecasting models as the most prevalent stand-alone forecasting models. However, no conclusive statement can be made regarding which of these approaches is more favourable, since these models have proven to outperform one another in several scenarios [28, 53, 67].
2. Recent research published after 2017 has indicated a strong shift away from shallow ML-based models, towards a focus on DL models. The rapid adoption of DL models is attributed to the increased PV system data set sizes and the availability of more input features. The ability of DL models to make more complex associations given the increased data availability has made DL models an obvious choice towards any further progress within the research field.
3. Due to the enhanced feature extracting capabilities of DL models, recent literature has further indicated a shift towards multi-step forecast models, also referred to as multi-target or multi-output regression models. This progression in the research field has also been reported on in the extensive literature reviews of A. Mellit *et al.* [18] and R. Ahmed *et al.* [19].
4. Recent studies still report on notable inconsistencies regarding forecasting accuracies between clear and cloudy days. Thus, accurate forecasting remains a challenging task for days with high intermittence due to cloud cover. Therefore, in an attempt towards further transparency, it has become the norm to publish results according to day-type, which at minimum could be classified as clear, cloudy and overcast.

1.3 Research motivation

From the thorough assessment of existing literature conducted, it is further revealed that the majority of existing research has been performed on small-scale (< 100 kW) PV systems. From the abundant collection of published literature, only five studies [30, 50, 77–79] could be found, which developed forecast models for large (≥ 20 MW) PV power plants. Evidently, research executed on utility-scale PV systems has indeed proven to be a rarity, resulting in a clear disconnect between published research and utility-scale PV power production.

Further evident from literature is that researchers generally develop forecasting models, based on a macro-level approach. With this conventional approach, models are trained on the global power output data as delivered by the PV system. Thus, a single forecasting model is developed with the intent to emulate the behaviour of the PV system as a whole. This is conceptually illustrated with Figure 1.4.

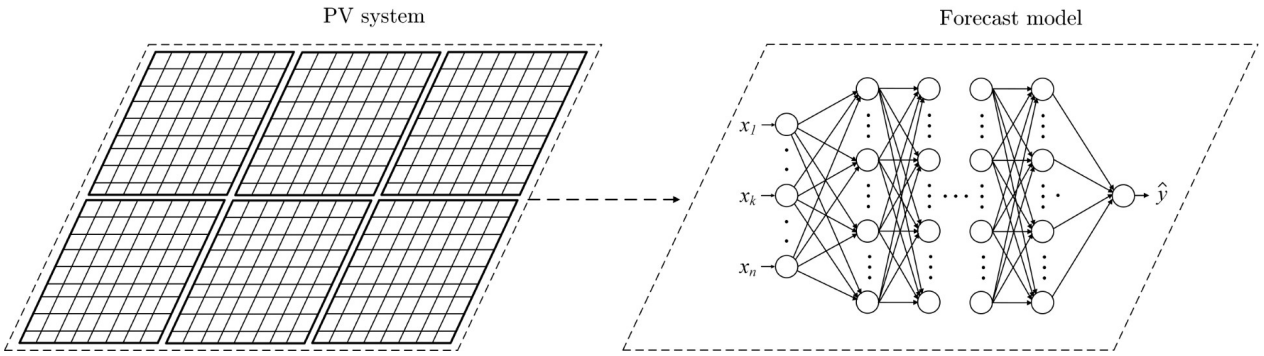


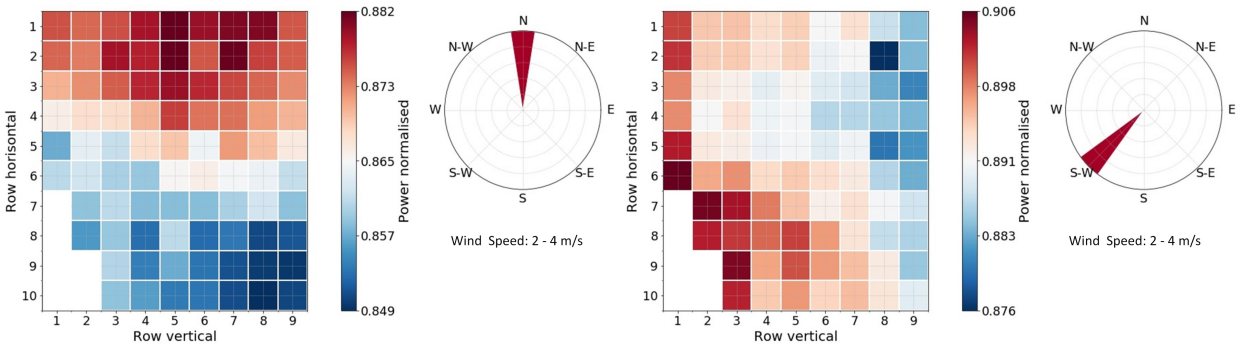
Figure 1.4: A conventional macro-level PV forecasting approach.

Evidence from literature suggests that PV-forecasting models based on this traditional macro-level approach, might disregard important low-level information, resulting in less accurate forecasts. To substantiate this statement, we firstly consider the physical size of large grid-integrated PV systems, which can equate to hundreds of thousands of square meters [m^2]. With such tremendous ground coverage by these utility-scale PV systems, a variation in power output can be anticipated for different localised segments within the system. PV installations are complicated, often consisting of several arrays, which are impacted by variations in topography, shading, orientation, dust soiling and degradation. Another obvious example is the aerodynamic influence of increased PV-system size, which can divert or restrict inter-system wind movement. This in turn leads to non-uniform air circulation, resulting in an uneven temperature distribution amongst modules, which directly impacts module performance [85–88].

As proof of the aforementioned claims regarding increased inter-system variability, due to increased PV-system size, a preliminary analysis is conducted on the data obtained for this research. This data was acquired from a commercial 75 MW rated PV-system, as presented with Figure 1.5.

Figure 1.5: Aerial view of the 75 MW PV-system used for the execution of this research [89].

The non-homogenous inverter power output behaviour due to wind movement, as captured for a moment in time, is demonstrated in Figure 1.6. Evidently, inter-system power output variations of up to 3 % occur during these scenarios of wind direction, given the relative position of the PV modules to one another.



(a) Power dynamics: North wind

(b) Power dynamics: South-West wind

Figure 1.6: Demonstration of the non-uniform effects of wind on the power output dynamics of the PV-system inverters, with one inverter block represented by a single square.

Importantly, the influence of wind on PV system power output dynamics, as illustrated in Figure 1.6, does not infer that wind has a dominant influence on PV system behaviour. It merely serves to prove the non-uniform exposure of PV system segments to the environment. Another potential source that can contribute to the increased variability of low-level power output dynamics is non-uniform, inter-module shading. Within large PV systems this is a common occurrence when sun altitude is low (early morning, late afternoon), so that longer shadows are projected onto adjacent PV module rows. However, an uneven PV-system surface topology (hills, mountains) will further amplify the non-uniform effects of shading and wind circulation, as for example seen with the 100 MW Les Mées solar farm [90] demonstrated in Figure 1.7.

Figure 1.7: Les Mées PV system with a non-uniform (rolling-hills) ground surface topology.

1.4 Research aims

Given the context of non-uniform power output dynamics and the limited research available on utility-scale PV forecasting, a crucial knowledge gap persists. Only to be further widened with newly commissioned PV systems rapidly expanding far beyond several MWs of installation size, as evident from the REN21 report [1]. Therefore, this knowledge gap concerning the effects of non-uniform, low-level power output dynamics on forecasting accuracy has become more relevant than ever. Given the fierce competition and immense pressure of energy market bidding, further improvements in utility-scale forecasting are crucial in the race towards increased forecasting accuracy. This has lead to the following research question, visually aided by Figure 1.8:

Is it possible to obtain an enhanced forecast model sensitivity to non-uniform power output dynamics, by using an aggregate of low-level forecasts?

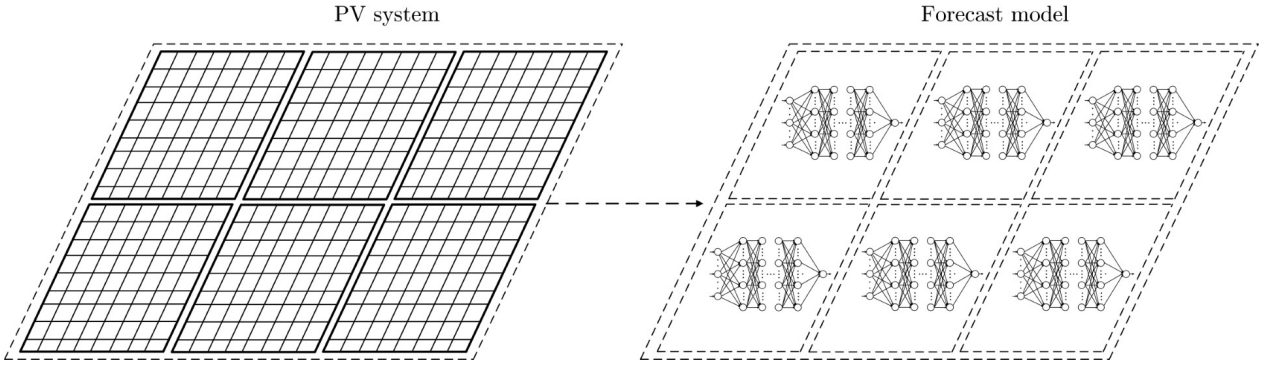


Figure 1.8: Suggested approach of aggregated low-level PV-system forecasts.

The closest example of research, which has partially touched on this question is found in the previously mentioned work of Z. Li *et al.* [61]. These authors presented inverter-level forecasts for a 6 MW rated PV system. With predictions delivered up to 1 h ahead, the inverter-level forecasts made a marginal improvement over the macro-level predictions. However, with the research executed on such a small scale (compared to $\geq$ 20 MW systems) with conventional ML-based models, the presented research question remains unanswered for large multi-megawatt PV systems.

In response to the main research question, it is crucial to consider the progressive shift of this research field towards the use of DL models. Given the increased ability of DL models (compared to widely used shallow ML models) to extract non-linear features and establish complex relationships, there is a possibility that DL models might prove powerful enough, so that aggregated low-level forecasts are redundant.

Therefore, secondary to the main research question:

With state-of-the-art deep learning models trained as a macro-level forecasting solution, what is the ability of these more powerful models to capture low-level, non-uniform PV-system power output dynamics?

1.5 Research hypothesis

It is hypothesised that the non-uniform exposure of large PV systems to heterogeneous environmental variables, is unaccounted for with the required detail by a traditional macro-level forecasting methodology. Therefore, it is anticipated that an enhanced forecasting accuracy can be obtained from aggregated low-level forecasts, which account for the non-uniform power output dynamics of utility-scale PV systems.

1.6 Research objectives

Based on the research questions and hypothesis, two principal objectives are established, each with an additional subset of objectives. This collectively serves as guideline for the successful execution of the presented research.

1 PV power forecasts are to be delivered for a real-world scenario, which accounts for the environmental, atmospheric and operational influences, together with the corresponding PV-system power output dynamics.

- Acquire raw, on-site measured weather data and power output data from a large multi-megawatt grid-connected PV system.
- The raw data must be processed so that a data set, without anomalies and erroneous data entries is obtained, representative of characteristic PV-system behaviour.

2 Execute the novel evaluation of an aggregated low-level forecasting methodology, compared to a conventional macro-level PV forecasting methodology.

- Deliver short-term (1 - 6 h ahead) PV power forecasts, with state-of-the-art DL models.
- Train and apply models with a traditional macro-level forecasting approach.
- Train and apply models to deliver forecasts for each inverter, which serves as the aggregated low-level forecasts.
- Develop and apply a systematic approach to ensure that all models are developed with an equal amount of effort, to avoid any evaluation bias.
- Ensure that the test data set used for evaluation is representative of a full year of power delivery, to avoid sample and confirmation bias.

1.7 Research scope and limitations

With the research objectives defined and context given regarding the fundamentals of PV forecasting as research field, this section articulates the boundaries of the work presented.

1.7.1 Forecast horison

With the intended aim of this research to assist with important fast-response scenarios, such as load balancing and energy bidding, a 1 - 6 h ahead forecast horison is ideal for these scenarios [8, 71]. These 1 - 6 h ahead predictions are to be provided at 15 min resolution, which equates to a total of 21 prediction points delivered.

It is decided that forecasts are not to be delivered for a time period less than 1 h ahead. Reason being that literature has proven the success of statistical and persistence models for intra-hour forecasts, making the use of DL-based models redundant for this time horison. Also, with a 45 min cut-off time for unit bidding on some of the energy exchanges like the EEX [7], there is no incentive to deliver forecasts for a sub 1 h horison, given the intended aim of the predictions.

Day-ahead forecasts exceeding 24 h and beyond are further not considered for this research, since this forecast horison has a different purpose as discussed in Section 1.2.2. Also, literature has indicated that day-ahead models benefit from weather forecasts as additional input, which are not to be considered for this research.

1.7.2 Input and output data

The forecasts to be delivered are based on a purely historic input data (power, weather data) set obtained from on-site measurements only. Data obtained from weather forecasts is not to be considered as input for the forecasting models. This allows for an independence on third-party weather predictions, which are not equally available at a high resolution for utility-scale PV-systems, often located in remote areas. Also, by staying independent from forecast data inputs, the research focus is maintained, which is to determine the ability of state-of-the-art models to capture and predict low-level PV-system behaviour. With the use of only historic data, the true stand-alone ability of the models applied can be revealed. This approach also allows for the reproducibility of the research presented, making the research more relevant to a large segment of the PV-forecasting community.

The forecasts to be delivered are based on a purely historic input data (power, weather data) set obtained from on-site measurements only. Data obtained from weather forecasts is not to be considered as input for the forecasting models. This allows for an independence on third-party weather predictions, which are not equally available at a high resolution for utility-scale PV-systems, often located in remote areas. With the use of only historic data, the true stand-alone ability of the models applied can be revealed. This approach also allows for the reproducibility of the research presented, making the research more relevant to a large segment of the PV-forecasting community.

1.7.3 Model selection

With forecasts to be delivered 1 - 6 h ahead, literature has proven that ML-based and DL-based models are ideal for this forecast horison. The research focus is on state-of-the-art models, with a proven track record. Therefore, based on recent research publications, forecasts are to be delivered with the use of FFNNs, LSTM-RNNs and GRU-RNNs. By focusing on these models, the research appeals to and delivers further value to a large subset of the PV-forecasting community.

From literature, it is important to note that not a single method is superior to all other methods, which can be applied to data from any location, under any circumstance. Therefore, the focus of this dissertation remains on the methodology of low-level forecasting, which can be universally applied, rather than the development and explicit claim of superior performance of a single model.

Also mentioned, forecast models can further be categorised as either deterministic or probabilistic. Following the detailed discussion presented in Section 1.2.5, it is mentioned that research focus will be directed towards deterministic models. These models have also been and remain to be the predominant focus of PV-forecasting literature. Since deterministic

forecasts can also be expanded to deliver probabilistic information, the use of purely probabilistic models will not be considered.

However, results presented are to be aided with uncertainty information to convey further insight into the effectiveness of the proposed inverter-level methodology.

1.7.4 Single-target vs. multi-target regression

With the goal of this research to deliver forecasts 1 - 6 h ahead, at 15 min resolution, a multi-step (21 time steps) forecast solution is applied. The reason for training Multi-Target Regression (MTR), or rather multi-output models, as opposed to a Single-Target Regression (STR) approach is primarily due to the increased representational power of DL models. This increased ability to capture complexity is also why recent literature has progressed towards MTR forecasting models [18, 71, 72].

Researchers have referred to the inherent advantage that ML-based MTR models have regarding the correlations between the multiple output values [91]. This advantage of MTR over STR models is especially emphasised by the review of H. Borchani *et al.* [91]. For STR models each model is trained for a particular prediction point in time, having no regard for the other output data points of the forecast horison. This also requires multiple models to be customised individually, which naturally makes the overall process of model training computationally expensive. However, with MTR each of the NN parameters (weights, bias values) are adjusted accordingly, within the context of optimising for the overall output of the model. Thus, model parameters are adjusted based on all of the relationships between the multiple output variables.

Finally, the recent reviews published by A. Mellit *et al.* [18] and R. Ahmed *et al.* [19] clearly outlines the movement of new research towards MTR forecasting. The recent work of H. Zhou *et al.* [49] also suggested the expansion of their research towards multi-step forecasting. For these reasons a MTR approach is adopted for this research.

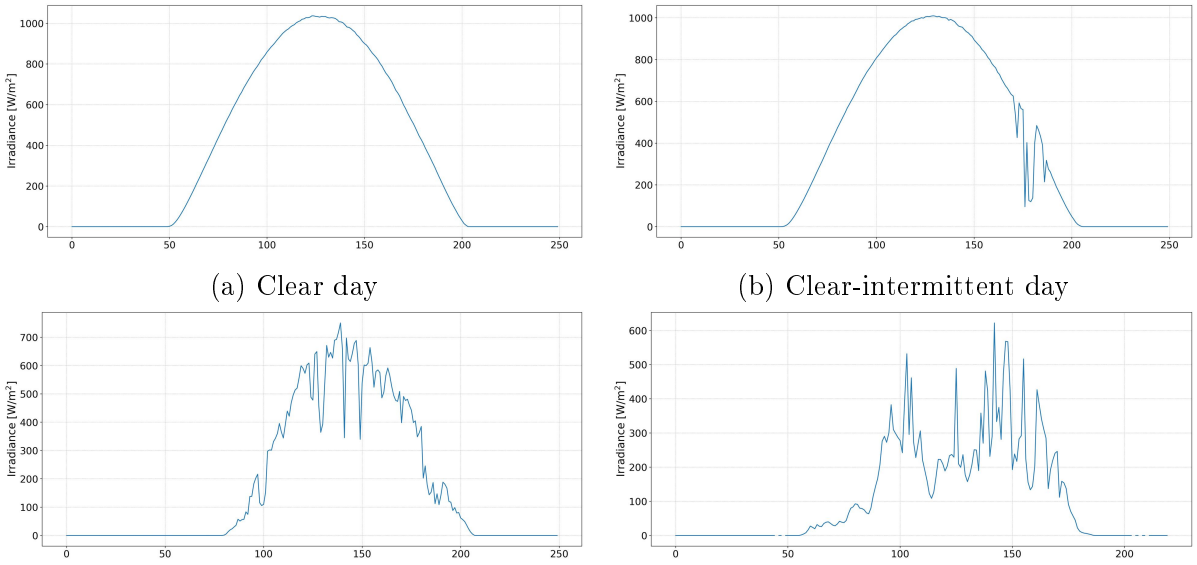
1.7.5 Day-type and seasonal forecasting

Individual forecast models can be specifically trained for each day type, characterised for example as clear, clear-intermittent, intermittent and cloudy, as illustrated by Figure 1.9. For this scenario, the primary data set is to be divided into subsets for each of these day types. Given that there is enough data available for each day type, this might be a feasible strategy. However, the concern with this approach is that it might not always be evident which day-type model to deploy. Especially since weather forecast services are not considered for this research, but only historical data.

In terms of a real-world commercial application, one could deploy all four day-type specific models simultaneously as a competitive ensemble approach [92]. Therefore, selecting which model to prioritise for the following forecasts, as the day continues.

For this research, models are developed to collectively forecast power output for all day types and all seasons. The models are also developed with the aim of being dynamically applied in a real-world forecasting scenario, which only requires models to be updated annually. It is not within the scope of this research to develop and compare models, which are trained and updated monthly or seasonally or for any other time period less than a year.

As mentioned, an objective of the research is to identify the degree at which macro-level forecasting models can capture non-uniform, low-level PV-system behaviour. Thus, developing models capable of adapting to all day types and seasons serves as a measure of generality,



(a) Clear day

(b) Clear-intermittent day

(c) Intermittent day

(d) Overcast day

Figure 1.9: Illustration of four different day types as determined by received irradiance.

demonstrating model ability to emulate low-level PV-system behaviour. However, it remains an objective to evaluate model performance for each specific day type.

1.8 Thesis overview

The remainder of the dissertation is structured as follows:

- Chapter 2: This chapter starts with an overview of the PV-system details and the location of the system. Then, with raw data recorded over a period of four years, the entire data set and process of data acquisition is presented. This chapter proceeds to elaborate on the extensive task of data-processing applied to the raw data, to obtain a final data set used for the development of the forecast models.
- Chapter 3: In this chapter a basic introduction of ML fundamentals and the theory regarding the FFNN, LSTM-RNN and GRU-RNN models is provided. This is followed by the development and application of a strategic framework for training models. The primary aim of this development framework being the ability to formalise the highly iterative and sometimes intuitive process of model development. A unique inverter-clustering approach is also introduced, which serves to deliver custom forecast models for each inverter, at a reduced computational cost without sacrificing accuracy. After this chapter, readers should have full confidence in the strategy applied to obtain unbiased macro-level and inverter-level forecasting models.
- Chapter 4: The results obtained for the macro-level and aggregated inverter-level forecasts are presented in this chapter. These forecasting results are also discussed, with the macro-level and inverter-level models directly compared to one another. A final verdict is delivered regarding the ability of DL-based macro-level models to capture low-level power output dynamics, in comparison to the proposed inverter-level forecast results.
- Chapter 5: This chapter serves to conclude on the research presented, with a brief summary of the results and findings.

Chapter 2

Data Processing

2.1 Introduction and overview

The focus of this chapter is on data acquisition, pre-processing and feature engineering. Firstly, technical details about the PV system are provided, together with the available data sensors from which data is acquired. With the raw data set introduced, the detailed step-by-step process applied to prepare the data for the models is revealed. With a clean data set established, representative of true PV-system behaviour, the selection of data is discussed and how this data is adapted to ensure that models are trained effectively. At the end of this chapter, readers should have full confidence in the final data set used to train and validate the models.

2.2 PV-system location and environment

The data used for the execution of this research is obtained from a utility-scale PV system, with a direct current (DC) rated capacity of 75 MW. As illustrated in Figure 2.1, the PV system has a North-facing, fixed tilt (30°) topology.

Figure 2.1: Aerial view from the side of the 75 MW PV-system used for the execution of this research (*Photo courtesy: Kurt Krog*).

Power is delivered from 84 inverters, each rated at 880 kVA, with a range of 2400 - 3480 PV modules connected to each inverter. This equates to power delivered from more than 312 500 PV modules, which collectively covers an approximate area of 1 000 000 m^2 (100 hectares or 250 acres). With the ground-surface topology of the immediate environment relatively flat, together with PV modules equally spaced with more than 3 m of edge-to-edge distance between module rows, no inter-module shading occurs during the primary hours of power production.

The location (30.161° S, 24.132° E) of this PV-system is in the Northern Cape region of South Africa, as indicated on the map of Figure 2.2. With more than 2600 kWh/m^2 of annually received direct normal irradiance, this location is ideal for solar PV power production, which is why this region hosts the majority of utility-scale PV power systems in South Africa. The immediate environment and the PV system itself has a relatively flat ground-surface topology. Therefore, with PV modules equally spaced at an edge-to-edge distance of more than 3 m between module rows, no inter-module shading occurs during the primary hours of power production. Regarding the general weather conditions, daily temperatures during the summer months of December to February approximately average 32 °C. For the winter months of June to August daily and night time temperatures usually average at 16 °C and 2 °C, respectively. During the rain season, which occurs between the summer and autumn months of February to April, an estimated 300 mm of rainfall can be expected. Wind also has a near constant presence, due to a flat surrounding topology of the environment (as seen in Figure 2.1), with typical wind speeds averaging at speeds of 4 - 5.5 m/s. Finally, the relative humidity levels range between 36 - 55 %.

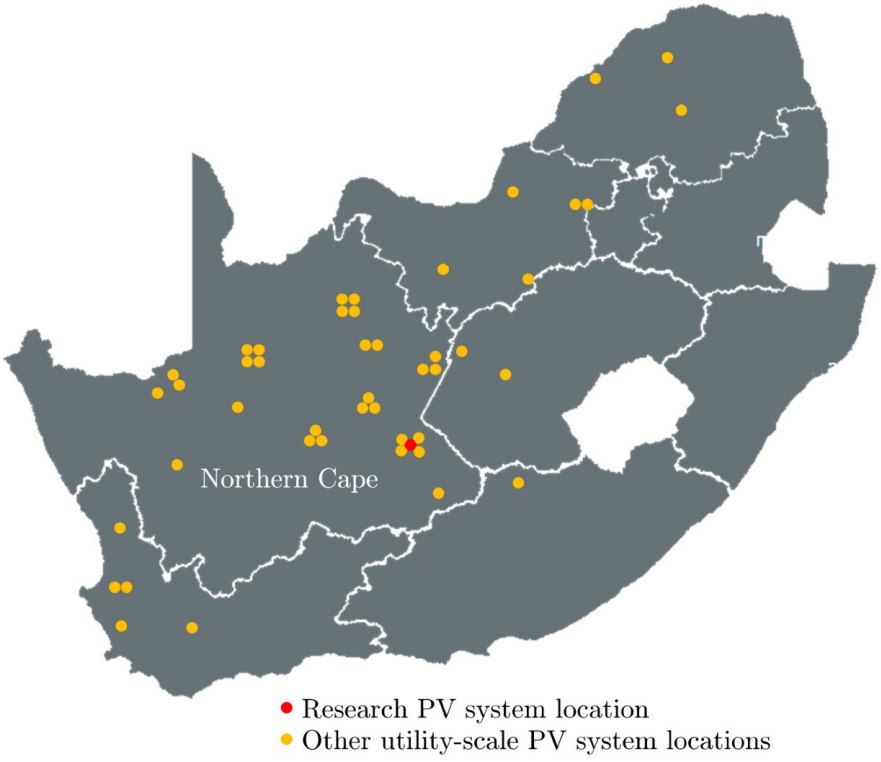


Figure 2.2: Location of PV system used for this research and the locations of other operational and planned utility-scale PV systems in South Africa [93].

Finally, with only a hand full of forecasting studies executed on stand-alone PV systems larger than 1 MW, this research delivers a real-world forecasting analysis of the largest

PV-system capacity to date.

2.3 Overview of data processing strategy

With the need to automate the process of cleaning raw data and simultaneously ensure that data-processing is applied with consistency to new incoming data, a data processing pipeline is established. Consisting of three stages, this data pipeline is demonstrated in Figure 2.3, with each stage focused on:

- Stage 1 - Raw data acquisition: Data retrieval & structuring
- Stage 2 - Raw data clean up: Anomaly detection, bad data elimination & replacement.
- Stage 3 - Final data pre-processing: Feature engineering & model data preparation.

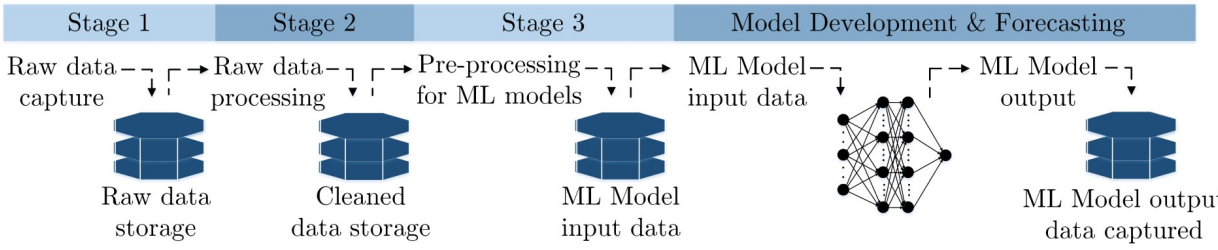


Figure 2.3: Three-stage data processing pipeline.

These three stages of data processing are applied with a principled and systematic data-science approach. The first goal is to obtain a truly representative data set, which accurately captures the PV-system dynamics in response to the environmental and atmospheric variables. This of course requires a data set that is free from anomalies, which could impair forecast model training. Secondly, the goal is to obtain a data set, representative of full cycles of seasonality, so that data-sample bias is avoided and to ensure that the forecasting models generalise well in terms of anticipating PV-system behaviour for all weather (daily and seasonal) variations.

2.4 Data pre-processing Stage 1: Raw data acquisition

2.4.1 Weather sensor data

Weather data is recorded at five different locations within the PV-system, as marked by the yellow circles in Figure 2.4. All of the weather stations are located within the PV system at a location and height representative of that of the PV modules. Only the control building weather station (indicated by CB) on the edge of the PV system, is placed at a height approximately double that of the other weather stations. These weather stations capture various weather-related variables, of which a full list is available as summarised in Table A.1 in Appendix A.1.

The weather data used as input for training the forecast models is presented in Table 2.1. It should be mentioned that only weather sensor CB measured absolute air pressure (AP), wind direction (WD) and relative humidity (RH). These input features in Table 2.1 are specifically selected, since this data can be easily obtained from standard on-site weather sensors.

By avoiding complex measurements and inputs such as an aerosol index, a light spectrum analysis, detailed back-plate temperature measurements, etc., the research presented has a higher degree of reproducibility by other researchers.

Table 2.1: PV system weather data used for model development.

Measurement	Unit
Global Horisontal Irradiance (GHI)	[W/m^2]
Ambient-temperature (T)	[°C]
Absolute-Air-Pressure (AP)	[hPa]
Wind direction (WD)	[0° − 360°]
Wind speed (WS)	[m/s]
Relative-Humidity (RH)	[%]

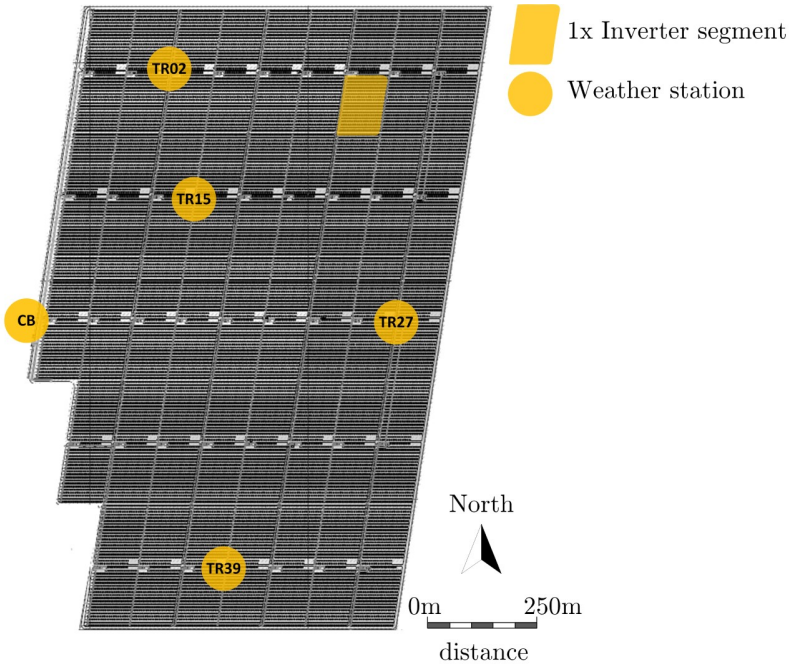


Figure 2.4: Weather station locations within the PV power plant.

2.4.2 PV-system data

Due to the large physical size of the 75 MW rated PV-system, data is captured at multiple strategic locations within the power plant. The available data sources, each of which captures various data points, are displayed in Table 2.2.

Table 2.2: Available PV-system data sources related to power output.

Data measurement devices	Number of devices
String monitors	840
Inverters	84
Low-voltage/Medium-voltage transformers	42
Medium-voltage/High-voltage transformers	2
Transducers	2

Figure 2.5 facilitates with a basic understanding of where these data sources are positioned within the PV system. From this layout it should further be clear that data is gathered at multiple levels of PV power production, defined as: string level, inverter-level and transformer level. Two power transducers measure the output from the high voltage transformers as delivered onto the electrical grid, which is the final point of measured power production.

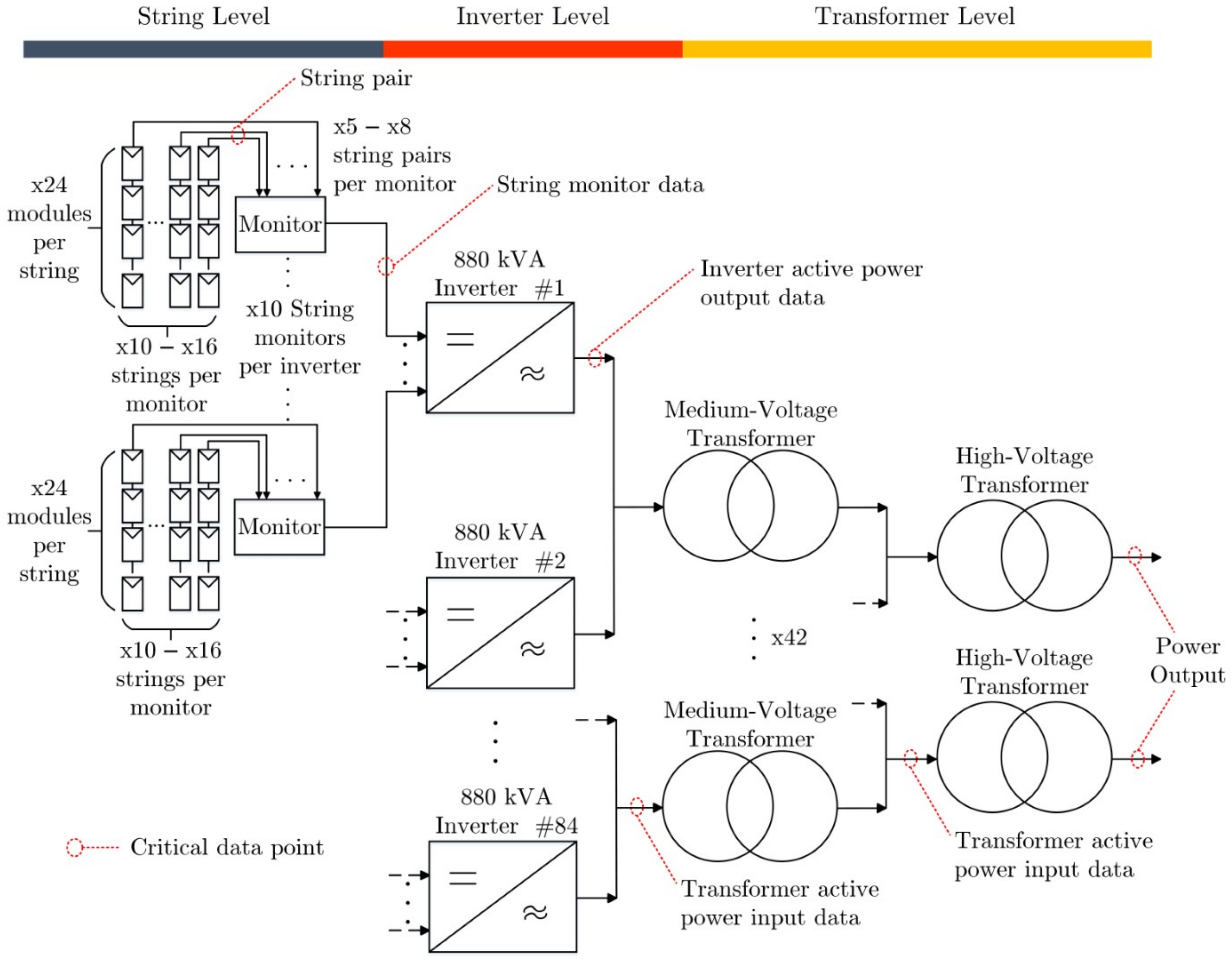


Figure 2.5: Low-level PV-system layout of the primary hardware together with important points of data capture.

2.4.2.1 String level and inverter level:

String level: As mentioned, the total PV system consists of 84 individual inverters. Each inverter has between 2400 - 3840 PV modules connected in string pairs, with each string consisting of 24 PV modules. This results in 100 - 160 strings for each inverter, depending on the configuration. The current supplied to each inverter is measured with a set of 10 string monitors, with each string monitor recording the data of 10 - 16 strings, as displayed in Figure 2.5. In total, the entire PV system consists of 13 021 strings. Ultimately, the data points captured by the string monitors represent the current [A] output from each of the string pairs, with one string pair consisting of two strings (of 24 PV modules).

Inverter Level: A considerable amount of data, as summarised in Table A.2 in Appendix A.2, is captured by each of the 84 inverters, individually rated at 880 kVA. Although there are evidently various data points, which offer valuable information, the most important data points with regards to the development of the forecast models are summarised in Table 2.3.

With the string-level and inverter-level power production demonstrated with Figure 2.5, the question arises regarding whether this research should not also consider aggregated string-level forecasts, instead of only inverter-level forecasts? With models trained for each string, there is a possibility to obtain an even higher sensitivity to low-level PV-system power output dynamics.

However, with more than 13 000 individual PV-module strings (~6500 string pair outputs measured), the computational expense required for the development of forecasting models, for each of these string pairs, is extremely high. Readers will have more context regarding the computational effort in light of the model development and results, presented in Sections 3.6.2 and 4.4, respectively. Also, considering that more than 6500 forecasts would have to be generated with each run, the real-time deployment of such an aggregated forecast is also anticipated to be unrealistic for a practical, real-world forecasting scenario. This would not only require forecasts to be delivered, but also the real-time processing of the string-level data set. The aim of the research outcome is to present a forecasting solution, which contributes towards the understanding of improving real-world forecasting. This requires that the research presented is both reproducible and scalable, further providing practical insights for the commercial PV-forecasting sector. Also, the novelty of the research remains at aggregated inverter-level forecasting, which is a knowledge gap still to be addressed in the PV-forecasting research field. For these reasons, string-level forecasts are not considered.

2.4.2.2 Transformer level and final output

As evident from Figure 2.5, power from the inverters are fed into the Low-voltage-to-medium-voltage (LV/MV) transformers. With the output of two inverters connected to one LV/MV transformer, there are 42 LV/MV transformers. These 42 LV/MV transformers are then divided into two groups, which feeds into two medium-voltage-to-high-voltage (MV/HV) transformers, individually rated at 40 MVA. The measured data points for each of the transformer types are displayed in Table A.3, presented in Appendix A.2.

The final stage of PV-system output is captured after the output of the two HV/MV transformers with a power transducer. The power transducer captures important AC related power output values. These captured data points are all summarised in Table A.4, presented in Appendix A.2. Finally, the most important data points captured, which are actively used for developing the power forecasting models, are summarised in Table 2.3.

Table 2.3: PV-system power data points recorded and used for model development.

Measurement	Unit	Data source	Description
Total AC Active power output	[W]	Inverter	The actual AC power output delivered by the inverter.
Active power target value	[W]	Inverter	The desired AC power target value set point, which is 880 kW by default. Value changes when curtailment is applied. Data point is used as a flag to identify when PV-system output was restricted.
Active Power Feed In	[W]	HV & MV transformer sensor data	Power as delivered from the inverters to each of the MV transformers. Data of HV transformer records power received from the MV transformers.
Total Active power	[kW]	Power Transducer	The actual AC power output delivered by the PV system as a whole onto the grid.

2.4.3 Data acquisition and storage

The data points recorded for the PV-system are logged by means of a Supervisory Control And Data Acquisition (SCADA) system, which allows data to be extracted in real time from the active PV-system. For this study, four years of available data is used for the execution of the proposed research. This data set consists of data points recorded at 1 min intervals for the period of 2015-05-01 to 2019-05-31 (49 months), which is received as a 560 GB collection of comma-separated values (CSV) files. With a unique tag-list item, it is possible to distinguish between the various measurements recorded on all of the devices.

For more details regarding the deployment of the database and the interaction with the obtained data during the process of model development, further information is provided in Appendix A.3.

2.5 Data pre-processing stage 2: Raw data clean up

2.5.1 Bad data elimination

With the structured raw database established the next step is to ensure that improbable data is eliminated. As for example, a faulty wind speed of 600 m/s or an irradiance measurement of -2147483648 W/m^2. Extreme values such as these occur for numerous reasons, such as bad data logging, communication issues, or due to bit overflow (for example a signed 32-bit overflow $+2147483647$, -2147483648). An example of these extreme values is illustrated in the raw temperature data as displayed in Figure 2.6.

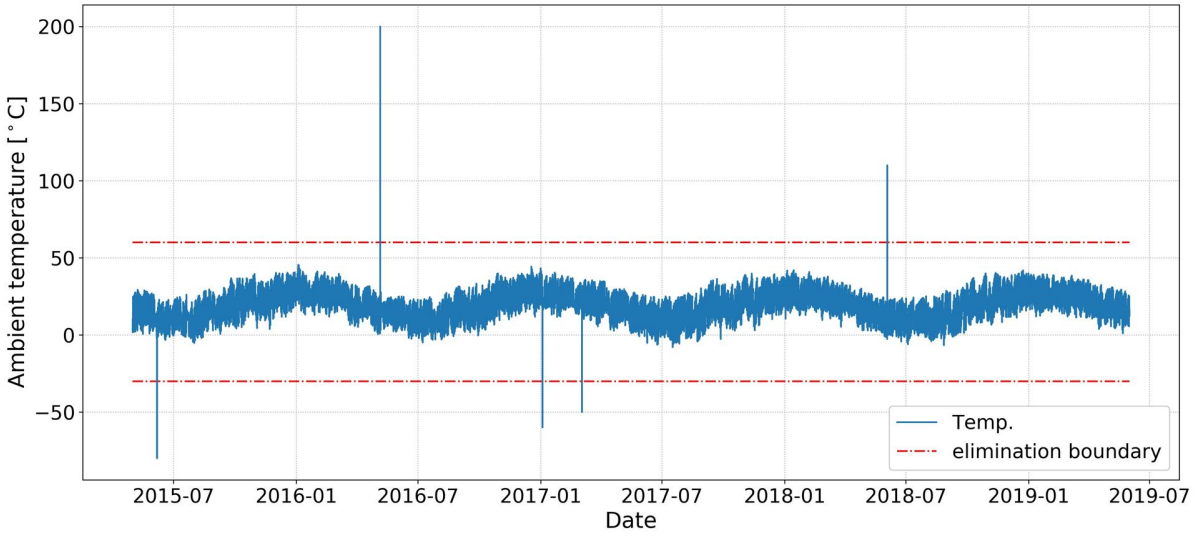


Figure 2.6: Ambient temperature boundary condition elimination.

Boundary conditions for each recorded data point are therefore established to streamline the process of bad data detection and elimination. The boundary conditions for the data points are established after careful considerations and recommendations from the local PV-system operators. Great care is taken to avoid eliminating data point anomalies, which could occur under extreme circumstances, such as an irradiance measurement of 1600 W/m^2 for example, or a wind gust of 20 m/s. A script is ran to identify all recorded values that fall outside of these boundary conditions, which then replaces these entries with NaN (Not A Number) values.

2.5.2 Data interpolation

With the raw data set cleaned from anomalies and unwanted data segments, there is still the challenge of missing data entries. With thousands of data entries made each day over a four year period, missing data points are a certainty for any utility-scale PV-system database. There are various causes for missing data, such as faulty sensors, communication errors or logging errors on the SCADA server side for example.

To avoid an unnecessary loss of data measurements, missing data points can be interpolated (replaced) with values based on the most recent historic data points recorded. An example of missing data points scattered between the longer continuous measurements is illustrated in the top graph of Figure 2.7. By not replacing these small segments of missing data will result in a discontinuous data set entry for an entire day of captured measurements. However, by interpolating these small missing data segments, data set continuity is maintained.

An extensive analysis, summarised in Appendix A.4, is presented regarding the type of interpolation (linear, polynomial) and the maximum number of allowed consecutive missing data entries, which may be interpolated. It was found that linear interpolation applied to missing data segments of less than 1 h resulted in a realistic replacement of data, which does not alter the intrinsic behaviour of the measured time-series data points. The bottom graph of Figure 2.7 illustrates this process of interpolation, where the missing data segments have all been interpolated.

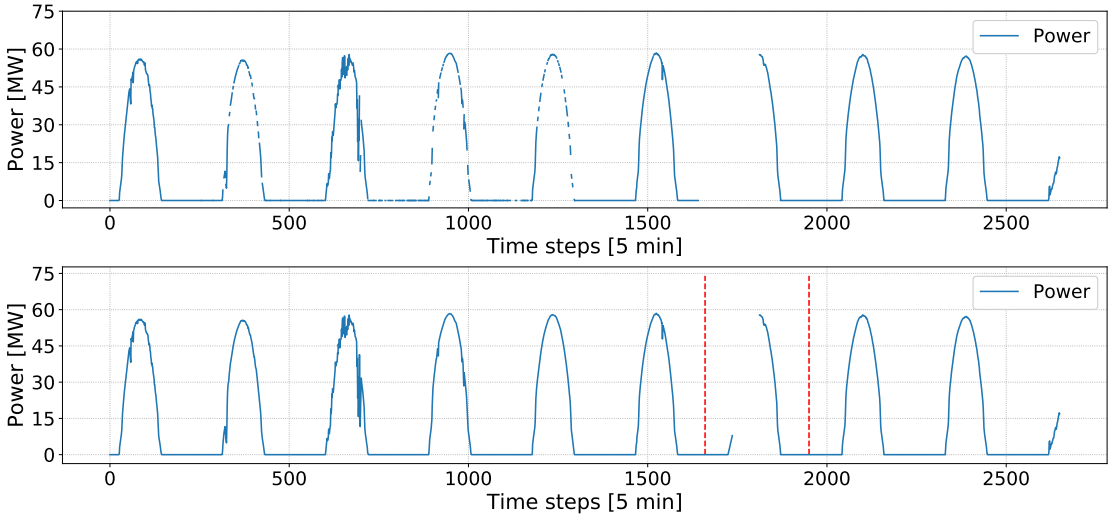


Figure 2.7: Example of interpolation applied to missing power data.

Only the segment, which represents a full day of recorded data, as indicated between the two vertical lines of Figure 2.7, could not be interpolated. Therefore, all data entries associated with this particular day are to be disregarded for the training of the models. The reason for dropping the entire day is due to the manner in which data is prepared for the models, which makes use of a sliding-window (look-back window) approach, as discussed in Section 3.4.1.

2.5.3 Data imputation and outlier elimination

With a large number of sensors constantly measuring data, sensor failure is a certainty for any utility-scale PV system. Given the data set obtained, typical reasons for prolonged segments (weeks, months) of missing data are attributed to:

- Faulty sensor data: This scenario occurs with sensor communication issues and bad database entries for example, which have gone unnoticed for a prolonged time period. Another commonality of prolonged missing data is related to deviating sensor measurements, attributed to a number of reasons. One reason being sensor damage or the necessity for recalibration. Since data (although faulty) is still retrieved from the devices, it is especially difficult to identify theses bad data entries. However, the advantage of having multiple sensors, which capture similar data, is the ability to identify faulty or deviating sensor measurements and potentially replace bad data segments.
- Missing sensor data: In this scenario, no data entries are made for a prolonged period of time, typically due to equipment malfunction. For example, a lightening strike damaged two inverters, which resulted in more than four months of missing data.

Unlike data interpolation, which utilises short-term historic data trends to replace data, imputation allows for the substitution of large segments of missing data, with another set of representative data points. As mentioned, it is crucial to train and evaluate models with a complete data set, representative of all seasons, to avoid sample and evaluation bias. Thus, replacing large segments of missing data is a necessity.

The execution of both the interpolation and imputation data processing strategies is presented in Appendix A.4. It should further be noted that all days where power curtailment was applied are also eliminated, as discussed in detail in Appendix A.4.4. Finally, a complete summary regarding all of the data set issues and how these issues were addressed, is presented in Appendix A.4.5.

2.6 Data pre-processing stage 3: Feature engineering and data preparation

2.6.1 Feature engineering

The goal of feature engineering is to obtain additional features, which can either be a unique representation (or transformation) of available data, or the creation of new input features. For this dissertation additional features are specifically created to give the models a context of time and the movement of the sun.

2.6.1.1 Solar position angles

Based on the strong correlation between irradiance and power output, domain knowledge is used to generate additional features regarding the movement of the sun. In particular, two angles referred to as the solar azimuth (Φ_S) and altitude (β) angles are used to specify sun movement, relative to the PV-system location (30.161° South, 24.132° East).

With Figure 2.8 as visual aid, the value of β is defined as the angular measure of the local horison (of the PV system), towards the geometric centre of the sun [94]. With the PV system located in the Southern hemisphere, the value of Φ_S describes the position of the sun, East or West of North as reference line. At solar noon, when the rays of the sun align perpendicular to a given line of longitude, then Φ_S is 0°. With the future position of the sun available from the β and Φ_S angles, the forecast models are given valuable input information. For a detailed description of how these angles are mathematically derived, so that the exact position of the Sun relative to the PV-system is obtained, please refer to Appendix B.

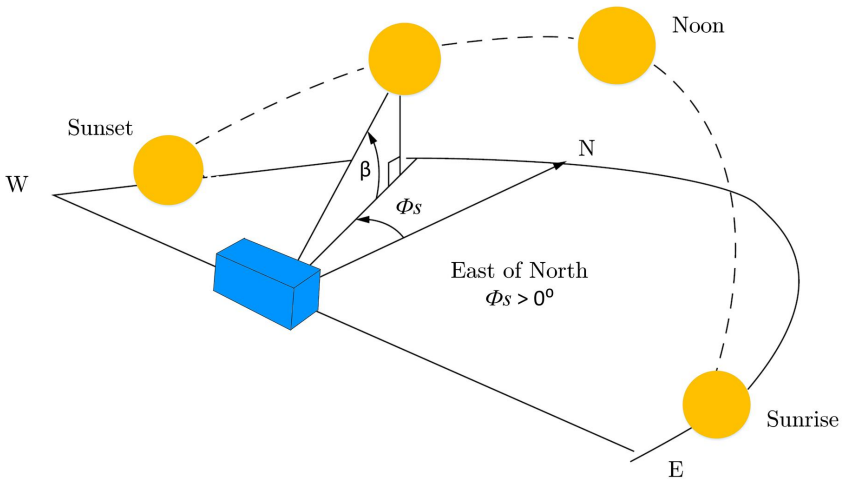


Figure 2.8: Illustration of the sun position as determined by the altitude β and azimuth Φ_S angles. *Source:* Redrawn illustration as presented by G. Masters [95, p.197]

.

The equations, used to determine the two solar angles as input features, are implemented as a Python module and were compared to well known Python libraries such as the pvlib-solarposition library [96], as well as on-line solar position tools [97]. From this the angles are determined for the exact forecast time of 1 - 6 h ahead. As will be demonstrated with Figure 3.8, a future-based sliding window is adopted to deliver this information as input for the models. From empirical analysis it is observed that with the additional input of the sun-position angles, model performance is greatly improved.

2.6.1.2 Additional time features

With GHI and PV-system power output highly correlated, there is an innate seasonality that can be seen in the normalised power output and GHI data of Figure 2.9. From the bottom graph there is an obvious pattern between power delivered and the hours from sunrise to sunset. To assist the training process of the models, this characteristic of time can be exploited. Therefore, the hour at which each forecast data point is to be delivered is extracted and provided as input feature for the ML models. Also, the month for which the forecasts are to be delivered is provided as input feature.

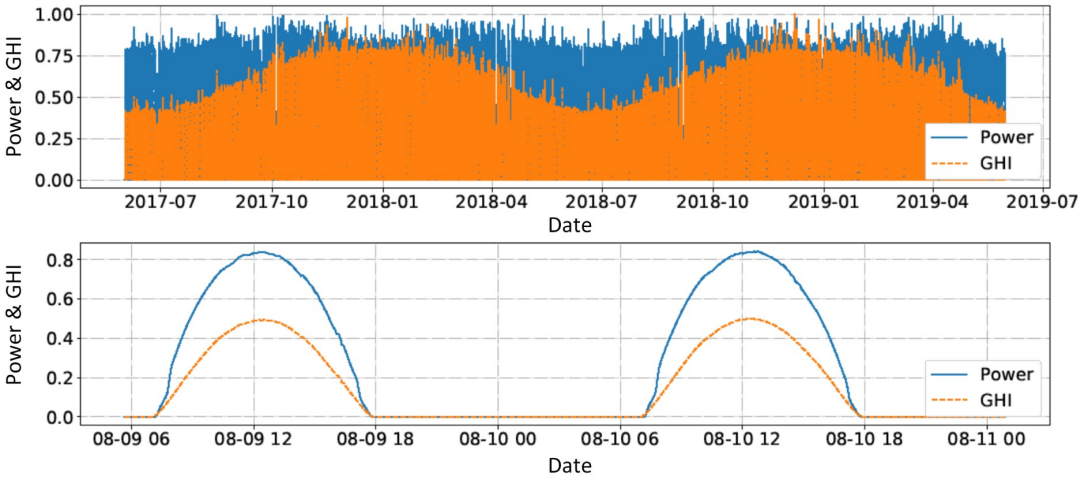


Figure 2.9: Illustration of annual and daily seasonality for the power output and GHI data.

Interestingly, from initial results with the use of hour of the day and month of the year as input features, it is found that the inclusion of the hour data did not contribute towards improved forecasting accuracy. The reason for this is attributed to the solar-position angles, which already embody enough time-based information, making the hourly data redundant. However, the use of the month of the year (applied as a one-hot encoded variable) proved to be valuable for improved model accuracy.

2.6.1.3 One-hot encoding

One-hot encoding [98, p.78] is used to further process data regarding the wind direction and month of the year. The reason for adopting one-hot encoding is due to the numeric values of these input features, which do not necessarily convey superiority. This data is categorised into N different possible categories, so that a binary feature vector array of length N is obtained.

For example, GHI as input feature has definite superiority regarding the numeric value since a higher value indicates that more irradiance is received, directly impacting the power output. However, wind direction does not have this association of superiority with the value of the wind direction measured in degrees. For example, Figure 2.10 displays the various wind-directions. In this scenario, a value of 359° is essentially the same as a wind direction of 1°. Thus, one-hot encoding is used to create a binary array with four elements, for each wind direction. The end result is demonstrated in Table 2.4. Initially, eight wind-directions were one-hot encoded, but analysis revealed four directions to be more than adequate.

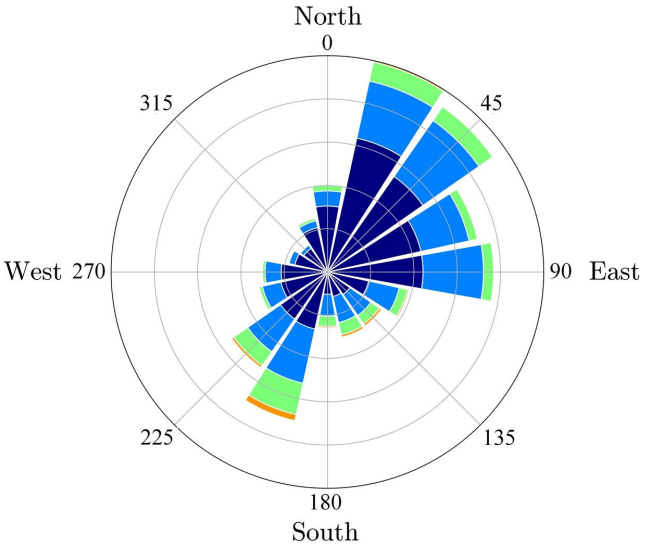


Figure 2.10: Wind direction and speed measured.

Table 2.4: One-hot encoding of measured wind direction values.

Wind direction		One-hot encoding			
Value [°]	Category	North	East	South	West
$315 < \text{val} < 360$	North	1	0	0	0
$0 \leq \text{val} \leq 45$	North	1	0	0	0
$45 < \text{val} \leq 135$	East	0	1	0	0
$135 < \text{val} \leq 225$	South	0	0	1	0
$225 < \text{val} \leq 315$	West	0	0	0	1

Input data concerning the month of the year is also one-hot encoded, since January (month number one) has more or less the same levels of received irradiance and temperatures as December (month number twelve). Several other one-hot encodings were also applied to other input features, such as hour of the day, seasons of the year and the solar position angles. However, none of these data adjustments delivered any notable improvements in model forecasting accuracies. Thus, only wind direction and month of the year is one-hot encoded.

2.6.2 Data preparation: Training, validation and test sets

With model development applied as a supervised training approach, the entire data set is partitioned into training, validation and testing data sets, as illustrated in Figure 2.11. With four years of available data, the training data consists of the first two years (2015-06-01 to 2017-05-31), with the data for the next two years allocated for the validation (2017-06-01 to 2018-05-31) and test (2018-06-01 to 2019-05-31) data sets (This data division approach is referred to as the fixed origin method).

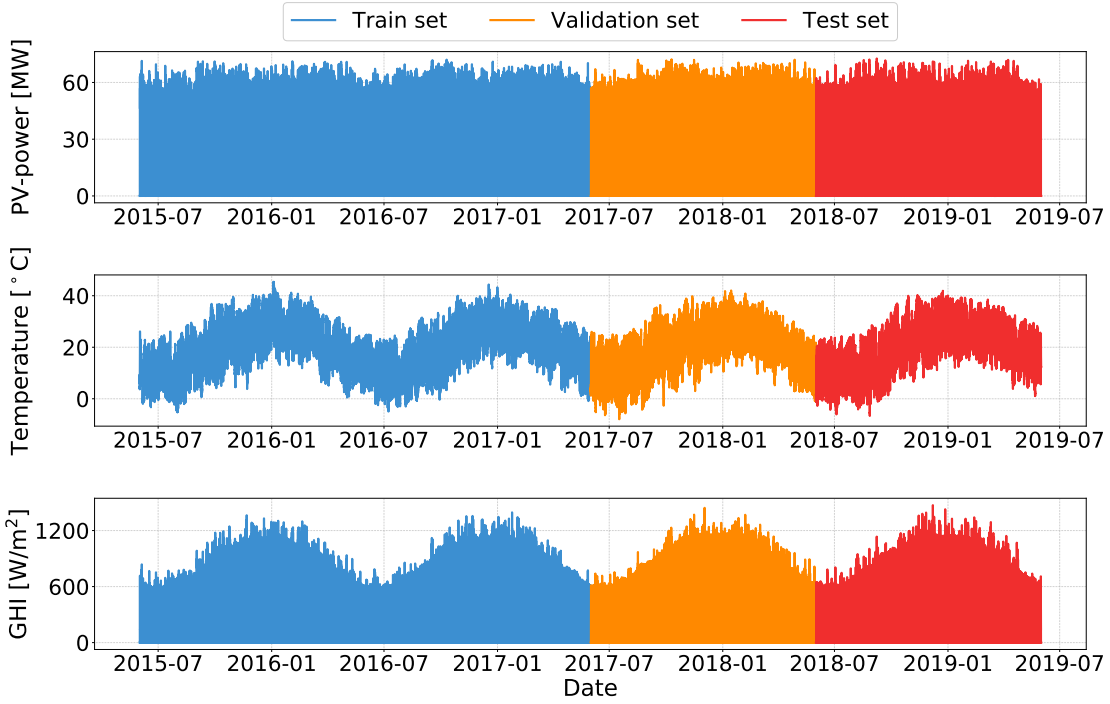


Figure 2.11: Partitioning of the training, validation and test data sets, with power, temperature and irradiance as examples.

The data allocations are made for these time periods, since it allows each data set to be representative of a minimum of one full annual cycle of seasonality (summer, autumn, etc.). This is important, as it was concluded from the literature review that the disparity, which exists regarding research claims of superior model performances, is partly attributed to scholars gauging model accuracies on incomplete test data sets. Thus, delivering sample-biased results that are not representative of all seasons, which do not necessarily generalise to all weather conditions during a year. Also, with the validation and test data sets each constrained to a single year of data, months and seasons are not over represented, so that another potential scenario of data bias is avoided.

Finally, it is important to mention that all decisions regarding model development and further optimisations are made according to the prediction accuracies obtained for the training and

validation sets. The test set is only used to deliver the final forecast model results, which allows for a fair and truly representative evaluation of obtained model forecasting accuracies.

This concludes the pre-processing of the received raw data set, which is now ready to be used for the development of the DL forecast models. Of course, further tailored adjustments of this data set are still to be made, to allow for the execution of the training strategy used for each unique forecasting model. More details regarding this are presented in Chapter 3, which is a discussion of the methodology concerning the process of training the models.

Chapter 3

Forecast model development

3.1 Overview

This chapter starts off with a brief overview regarding the basics of ML and the theory concerning the FFNN, LSTM-RNN and GRU-RNN models. This provides the necessary background to help understand the logical procession of the applied training methodology. The chapter proceeds to focus on the strategies applied towards model development. With two sets of forecasting models developed, namely PV-system macro-level and inverter-level models, a systematic approach is presented, which ensures that an unbiased amount of guided effort is applied to all processes of model development.

3.2 Feedforward Neural Networks

FFNN models, which are also referred to as Multi-layer Perceptrons (MLPs), have a proven track record of accomplishment as a PV-forecasting solution and remains a popular choice among researchers. Apart from the extensive PV power forecasting reviews published by U. Das *et al.* [15], D. van der Meer *et al.* [17], F. Barbieri *et al.* [2] and A. Mellit *et al.* [18], which confirm this statement, this is also true for the field of irradiance forecasting, as mentioned by M. Husein *et al.* [83]. Therefore, a FFNN model is selected to investigate the hypothesis that low-level forecasting should result in an improved forecasting ability, as compared to models trained to anticipate the global behaviour (macro-level) of the PV system.

3.2.1 Fundamental operation

The wide adoption of ANNs can be attributed to the ability of these models to establish complex and non-linear relationships between input and output variables, therefore acting as universal approximators [6, 25]. Similar to how the biological neuron serves as the processing element of the nervous system, so does the artificial neuron serve as processing element of an ANN [26]. Figure 3.1 illustrates a conceptual ANN framework consisting of the artificial neurons and the weighted connections between these neurons.

The basic operation of any FFNN starts with input data received by the input layer, which is then passed on towards the artificial neurons in the hidden layers. The computational results are then propagated forward from the hidden layers to the output layer, where a final computational process is executed and a response is provided. Information is passed from one neuron to the next by means of connections referred to as synapses, as indicated in Figure 3.1 [99, p.80]. Importantly, these connections between neurons each have a numeric

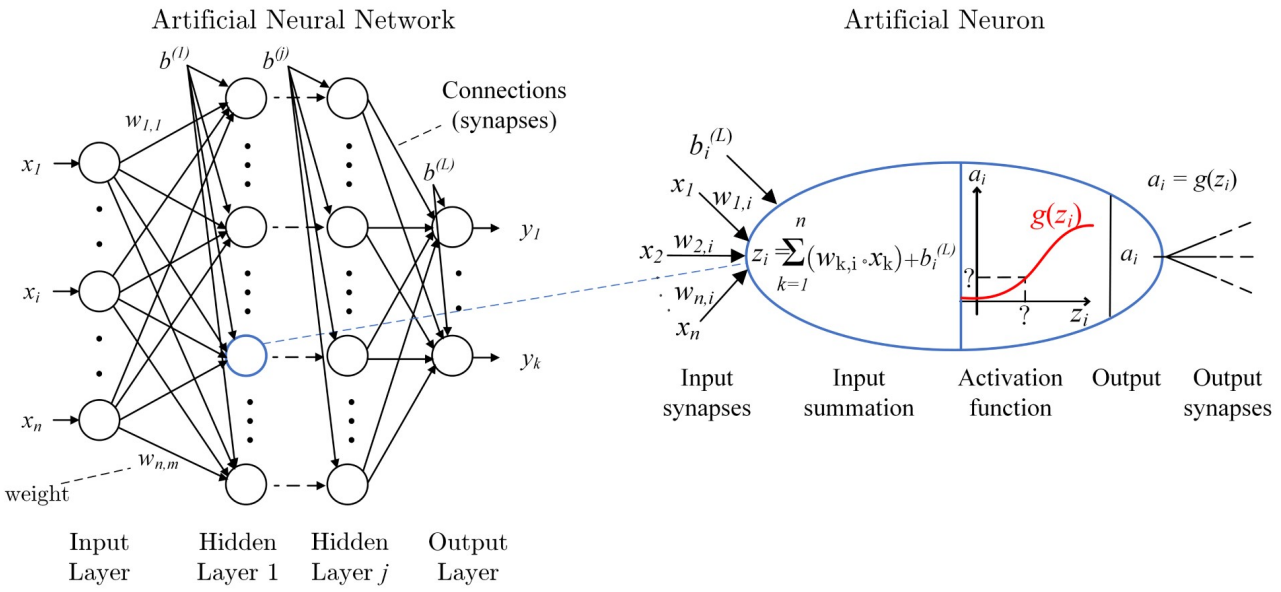


Figure 3.1: Illustration of the basic ANN structure and neuron.

weight ($w_{k,i}$), which determines the level of influence of the information received (x_k) from a particular neuron [100, p.8].

Effectively, the artificial neuron consists of two parts, namely the input summation $z_i^{(L)}$ and the activation function $g^{(L)}(\cdot)$ assigned to layer L. The input summation of a specific neuron i, in layer L, is defined by Eq. (3.1) of which the result is then delivered to and transformed by the activation function $g^{(L)}(\cdot)$ of the neuron. This is demonstrated with Eq. (3.2) [100, p.8].

$$z_i^{(L)} = \sum_{k=1}^{n} w_{k,i} \cdot x_k + b_i^{(L)} \tag{3.1}$$

$$a_i^{(L)} = g^{(L)}\left(z_i^{(L)}\right) \tag{3.2}$$

The notation of Eq. (3.1) and (3.2) is defined so that variable $w_{k,i}$ represents the weight of each neuron connection incoming from the previous layer, with k as the connection of hidden neuron k in the previous layer $(L-1)$ and hidden neuron i of the current layer L. The input values received by a neuron in layer L (from a previous neuron in layer $L-1$) is marked as x_k, with the total number of neurons in the previous layer represented by n. The bias term of layer L, which serves as input for neuron i is represented with $b_i^{(L)}$. Finally, with the neuron inputs collectively defined by $z_i^{(L)}$, the activation function$g^{(L)}(\cdot)$ is applied to $z_i^{(L)}$, with $a_i^{(L)}$ as the activation output value of the neuron.

With the aim of this research to develop deep FFNN models, with multiple hidden layers, Eq. (3.2) serves as the basic building block of calculating neuron output values and can be expanded to Eq. (3.3), which represents the output of neuron i in the next hidden layer $(L+1)$. The full mathematical expansion is demonstrated with Eq. (3.4).

$$a_i^{(L+1)} = g^{(L+1)}\left(z_i^{(L+1)}\right) \tag{3.3}$$

$$a_i^{(L+1)} = g^{(L+1)}\left\{\sum_{k=1}^{m} w_{k,i} \cdot g^{(L)}\left(\sum_{k=1}^{n} w_{k,i} \cdot x_k + b_i^{(L)}\right) + b_i^{(L+1)}\right\} \tag{3.4}$$

3.2.2 Activation function

As shown in the previous section, the output value ($a_i^{(L)}$) of a neuron in layer L is determined by the activation function $g^{(L)}(\cdot)$ applied to $z_i^{(L)}$. Several activation functions can be utilised for ANNs, which includes both linear and non-linear activations. Literature regarding PV forecasting, indicates that the most commonly used activation functions are Sigmoid, Hyperbolic Tangent Sigmoid and the Gaussian Radial Basis activation function [15, 19]. These and other popular activation functions are summarised in Table 3.1.

The Rectified Linear Unit (ReLU), defined by Eq. (3.5), is a popular activation function, proven to be very effective in other disciplines [101]. The main reason being that ReLU significantly improves the convergence speed of training algorithms, since it avoids the issue of vanishing gradients [101].

$$g(z) = \begin{cases} 0; z < 0 \\ max(0, z); z \geq 0 \end{cases} \tag{3.5}$$

Another version of this activation function is the Leaky Rectified Linear Unit (LReLU), which addresses the issue of stagnant weight updates, which can be an issue for the ReLU activation function. More recent advances made on the performance of these (LReLU, ReLU) activation functions are Exponential Linear Units (ELUs) [102] and Gaussian Error Linear Units (GELUs) [103]. Other variants further include Random-ReLU (RReLU), S-Shaped ReLU (SReLU) [104], Parametric Exponential Linear Unit (PELU) and Scaled Exponential Linear Units (SELU) [105]. However, as concluded by the comprehensive summary of C. Nwankpa [106], new DL architectures published still mostly rely on the ReLU activation function, outlining the fact that the more recent activation function variations on ReLU need more traction.

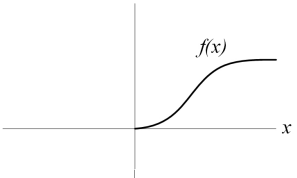
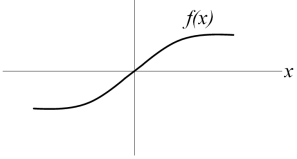
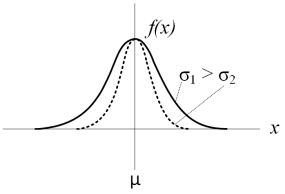
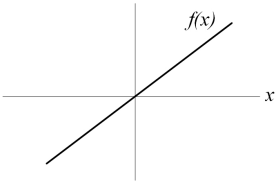
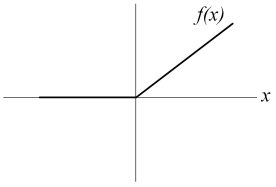
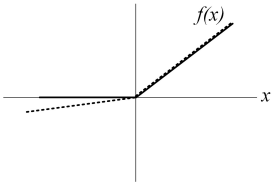
For this research ReLU is selected as activation function. Apart from the computational advantages, it is especially appropriate for PV forecasting, since predicted power values below zero are suppressed as zero.

3.2.3 FFNN model training and cost function

With the use of a Gradient-Descent (GD) optimiser, back-propagation and a cost function, the FFNN models are trained. Given the input feature vector $\boldsymbol{x}$, this information propagates through the FFNN, being delivered and transformed by each of the neurons, until a final output vector is delivered by the output layer as $\hat{\boldsymbol{y}}$.

The accuracy of the obtained FFNN output is then determined based on the difference between the output values $\hat{\boldsymbol{y}}$ and the target vector $\boldsymbol{y}$ values. The purpose of the cost function is to determine how aggressively errors should be penalised. Since the goal of forecasting power output is a regression task, typical cost functions that can be applied are MAE, Huber loss and Mean Squared Error (MSE) functions.

Table 3.1: Popular activation functions.

Activation function	Formula	Graphic illustration
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	
Hyperbolic tangent sigmoid	$f(x) = tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	
Gaussian radial basis	$f(x) = exp\left(\frac{-\lvert x-\mu\rvert^2}{2\sigma^2}\right)$	
Linear	$f(x) = x$	
ReLU	$f(x) = \begin{cases} 0, x < 0 \\ max(0, x), x \geq 0 \end{cases}$	
LReLU	$f(x) = \begin{cases} \alpha \cdot x, x < 0 \\ x, x \geq \theta \end{cases}$	

On a technical point, the loss function is actually what defines how aggressively errors are penalised. As presented with Eq. (3.6), the cost function $J(\cdot)$ is the average of all of the loss function calculations, as determined for each training example i, consisting of m different training samples. The $\boldsymbol{w}$ and $\boldsymbol{b}$ variables represent all of the weight and bias parameters of the model.

$$J(\boldsymbol{w}, \boldsymbol{b}) = \frac{1}{m} \sum_{i=1}^{m} \left(\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)}\right)^2 \tag{3.6}$$

For this research, the MSE is utilised as cost function, since outliers are penalised more aggressively. Considering that the adverse affects on the electrical grid are proportional to the size of the forecast error, the more aggressive MSE is ideal as the preferred cost function.

The objective of the GD optimiser is to adjust the weight ($w_{k,i}$) and bias ($b_i^{(L)}$) parameters accordingly, so that the smallest margin of error is obtained, and the cost function is minimised [107, p.370]. Finally the training process is halted when a set of $\boldsymbol{w}$ and $\boldsymbol{b}$ parameters is found, which produces either a solution that adheres to a desired accuracy, or the lowest possible error is found.

Now, given the solution space as defined by $J(\boldsymbol{w}, \boldsymbol{b})$, GD as optimisation algorithm adjusts the $\boldsymbol{w}, \boldsymbol{b}$ parameters in the direction defined by the gradient of the error made. Ideally, with each adjustment of the parameter, a smaller error is made. These iterative updates regarding the adjustments made to the parameters are represented by Eq. (3.7) and Eq. (3.8) as:

$$w_{k,i}^{(L)} := w_{k,i}^{(L)} - \alpha \frac{dJ(\boldsymbol{w}, \boldsymbol{b})}{dw_{k,i}^{(L)}} \tag{3.7}$$

$$b_i^{(L)} := b_i^{(L)} - \alpha \frac{dJ(\boldsymbol{w}, \boldsymbol{b})}{db_i^{(L)}} \tag{3.8}$$

The variable α in Eq. (3.7) and (3.8) is the learning rate, which determines how big the iterative steps should be to adjust the given parameter. From these two equations it should further be obvious that GD optimisation attempts to minimise the hypothesis function output $\hat{\boldsymbol{y}}$ by changing each of the $w_{k,i}^{(L)}$ and $b_i^{(L)}$ parameters. Evident from the GD updates made to the parameters, is that the gradient, as determined by the cost function, is required for each parameter. Reason being that in a FFNN with multiple layers the output of the FFNN, is no longer defined by the original input features $\boldsymbol{x}$. Instead the output is defined by several transformations of these input features, which occurs in every layer as the input features propagate through the FFNN.

Fortunately, with back-propagation the partial derivative of the cost function $J(\boldsymbol{w}, \boldsymbol{b})$ can be determined, with respect to each individual weight $w_{k,i}^{(L)}$ and bias $b_i^{(L)}$ parameter in the network, so that a gradient can be obtained and fed into the GD optimiser [108, p.197]. Thus, the fundamental idea behind back-propagation is effectively the chain rule of mathematics. With a vast amount of literature and resources available on the theory and application of the back-propagation algorithm, an in depth mathematical derivation of the back-propagation algorithm is considered redundant. The interested reader is referred to I. Goodfellow *et al.* [108, pp. 197-217], H. Kamper [109] and C. Olah [110] regarding the derivation of back-propagation, which have done well to effectively explain the concept.

3.3 Recurrent Neural Networks

For this research both LSTM-RNN and GRU-RNN models are used to develop the PV-forecasting models. Reason being the good track record of LSTM-RNN models in the PV-forecasting community, whereas GRU-RNN models are fairly new to the field, but have delivered competitive forecasting results [74]. Further motivation to explore the use of both models is due to the extensive empirical studies completed by J. Chung *et al.* [111] and R. Jozefowics [112], both of which could not nominate an overall superior time-series forecasting model.

3.3.1 Input data processing

A key limitation of FFNNs, which restricts the data set conditions for these models to perform well, is attributed to the way in which input data is provided to the model. FFNNs

require that the temporal dependence of the data is specified upfront, so that the entire sequence of data points is received at once [70, p.196]. As demonstrated in Figure 3.2, the time-steps representing the temporal dependence of the data, are flattened so that each time-step serves as an input feature to the FFNN model. This effectively results in a model, which has no temporal context regarding the order between the data observations [113, p.8]. Scaling also becomes an issue, because as the number of time-steps increase, so does the number of input features for the FFNN, which in turn increases model complexity.

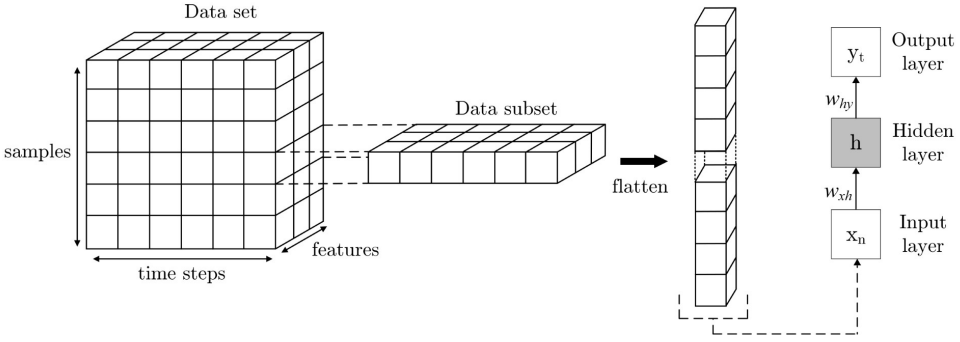


Figure 3.2: FFNN input data.

RNNs are specifically designed to process a sequence of data and therefore have the ability to scale to longer data sequences [108, p.363]. This ability enables RNNs to prioritise specific historic segments of data, which are information rich and relevant, together with the most recent data points. This is in contrast to a FFNN, which is equally impacted by data further down the past and the most recent data points. As illustrated by Figure 3.3, the subset of data to serve as input for the RNN model is not flattened out as in the case of the FFNN. Instead, each time-step together with the features associated with a particular time-step of the data subset, is sequentially delivered to the RNN model. It is due to this format and use of input data, that allows for an improved ability to scale to longer data sequences and thus account for more information upon delivering an output.

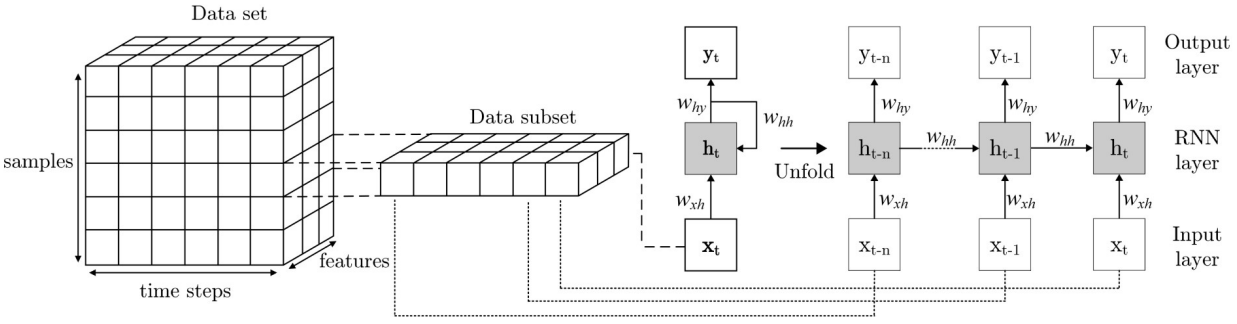


Figure 3.3: RNN input data.

3.3.2 RNN layer and training

Similar to the FFNNs, RNNs also map a vector of input variables $\boldsymbol{x}$ to an output target vector $\boldsymbol{y}$ by means of the assigned neuron weights, bias and activations. However, RNNs also have additional feedback connections, which allows the recurrent unit outputs to be fed back into itself as additional input [108, p.163]. These recurrent connections, as illustrated in Figure 3.4, allow the model to gain context regarding the past observations, so that information persists. This is evident from the recurrent neuron output $\boldsymbol{y}_{(t)}$, which is a

function of $\boldsymbol{x}_{(t)}$ and $\boldsymbol{y}_{(t-1)}$, each modified with a weight parameter (w) assignment, where the subscript (t) indicates the time-step within the sequence [114, p.384]. In turn, $\boldsymbol{y}_{(t-1)}$ is a function of $\boldsymbol{x}_{(t-1)}$ and $\boldsymbol{y}_{(t-2)}$, with the pattern repeating so that $\boldsymbol{y}_t$ is ultimately a function of the first input (at $t = 0$). This is why RNN models have the ability to prioritise the influence of data more recently seen, unlike a FFNN which is equally impacted by data further down the past, as well as the most recent data points

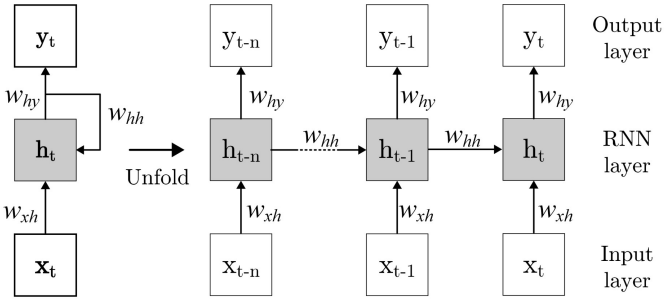


Figure 3.4: RNN cell unfolding. *Source:* Redrawn as presented by A. Géron [114, p.382]

3.3.3 RNN training

As mentioned in Section 3.2.3, back-propagation is used to obtained model parameter gradients, which are then used by the GD-based optimiser to update these parameters (weights and biases). With the conceptual explanation of RNN memory in Section 3.3.2, where RNN cells are unrolled through time for each of the input data time-steps, back-propagation is applied. This application of back-propagation in the context of training RNNs is referred to as Back-Propagation Through Time (BPTT) [108, p.374].

The difference between BPTT and ordinary backpropagation is simply the fact that the output sequence is now evaluated for each output of the individual time-steps of the RNN cell. Thus, errors are calculated and accumulated for the output of selected time-steps as $J\left(\boldsymbol{Y_{t-0}}, \boldsymbol{Y_t}, ..., \boldsymbol{Y_{t-max}}\right)$ [114, p.389]. In this context $\boldsymbol{Y_t}$ refers to the output of time-step t, given the assigned parameters, with $t - 0$ and $t - max$ as first and last time-step and $J(\cdot)$ as the cost function. The gradients of the cost function, are then backwards propagated through the unrolled network, so that the parameters can be updated with the gradients determined by BPTT [114, p.389]. Note that the same model parameters $\boldsymbol{w}$ and $\boldsymbol{b}$ are used for the unrolled time-steps of the RNN, which is why BPTT works.

With one time-step representing one input (input features), which delivers a single output, in similar fashion each time-step is rolled back and the gradients are calculated. With this process considered, BPTT can become very computationally expensive as the total number of time-steps are increased. Thus, for n time-steps as input, a derivative needs to be calculated n times to make a single parameter (weight, bias) update. So if n is very large (1000 time-steps for example), then model training will be very slow.

Finally, RNNs can be applied in a variety of different architectures, which is dependent on the sequence modelling objective of the model developed. For this research, the LSTM-RNN and GRU-RNN models are applied as a sequence-to-vector (many-to-one) architecture [114, p.383].

3.3.4 Long Short-Term Memory Recurrent Neural Network

LSTM-RNN networks are still considered to be a state-of-the-art sequence learning technique. From the background provided in Section 1.2, this is especially evident considering the number of recent research publications (years 2018 - 2020), utilising the LSTM-RNN model as primary forecasting technique.

LSTM-RNNs were first introduced by S. Hochreiter and J. Schmidhuber [115], with further contributions made by the well referenced works of F. Gers *et al.* [116] and A. Graves and J. Schmidhuber [117]. The limitation of RNNs is clearly outlined by S. Hochreiter and J. Schmidhuber *et al.* [115], where it is mentioned that RNNs are limited regarding the number of time-steps (long-term temporal dependency) that can be processed to effectively map input to output data. This is due to the recurrent connections, which cause the model weights to vanish or increase exponentially, leading to a vanishing and exploding gradients problem [118, 119]. Fortunately, this problem is solved by the self-connected gates of the LSTM hidden units, also referred to as memory cells [119]. These memory units are what make the LSTM-RNN different to a regular RNN.

The self-connected gates, as demonstrated in Figure 3.5, enable the LSTM-RNN model to have an increased sensitivity to the long-term dependencies of the time-series data, since data can now be written, read or removed from memory [119]. The LSTM memory cell gates are referred to as [115]:

- Input gate $\boldsymbol{i_t}$: Controls how the memory state is updated from the input.
- Forget gate $\boldsymbol{f_t}$: Determines the information to delete from memory.
- Output gate $\boldsymbol{o_t}$: Delivers the output based on the memory state and the received input data.

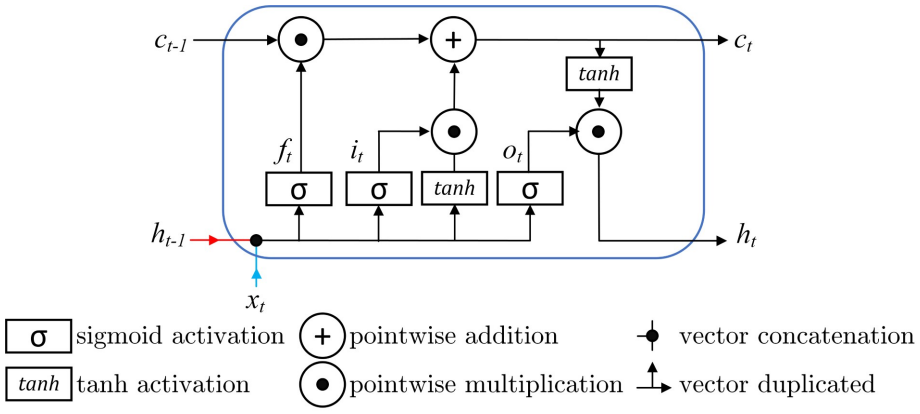


Figure 3.5: LSTM cell (*Source*: Image redrawn as presented by C. Olah [110]).

With reference to the LSTM-RNN memory unit in Figure 3.5, starting from bottom left is the forget gate vector $\boldsymbol{f_t}$, for which the output at time-step t is defined by Eq. (3.9) as:

$$\boldsymbol{f_t} = \sigma(\boldsymbol{W_{fh}} \cdot \boldsymbol{h_{t-1}} + \boldsymbol{W_{fx}} \cdot \boldsymbol{x_t} + \boldsymbol{b_f}) \tag{3.9}$$

From Eq. (3.9) $\boldsymbol{W_{fx}}$ and $\boldsymbol{W_{fh}}$ represent the forget gate weights, which consist of the parameter weight matrices assigned to the input vector $\boldsymbol{x_t}$ and the previous LSTM cell output

vector $\boldsymbol{h_{t-1}}$, respectively. The bias parameter values are represented by vector $\boldsymbol{b_f}$. Evidently, all of the forget gate values are scaled between a range of 0 to 1 by the sigmoid activation, which determines how much information is to be disregarded (forgotten).

For more detail, Eq. (3.10) represents the mathematical expansion of Eq. (3.9) of a single forget gate unit $f_i^{(t)}$ for time-step t of LSTM cell i [108, p.399]. The variable $w_{k,i}^{fh}$ represents the forget gate hidden state (super-script fh) weights assigned to the previous hidden-state output $h_i^{(t-1)}$ of cell k. The $w_{k,i}^{fx}$ variable is the weight parameters assigned to the input vector (superscript fx) values $x_i^{(t)}$, which is the i^{th} item of the current input vector ($\boldsymbol{x}_{(t)}$) for time-step t. The forget gate bias parameter of memory cell i is represented by b_i^f. Once again, the sigmoid activation function represented by σ, scales the particular forget gate unit $f_i^{(t)}$ to a range between 0 and 1, which determines how much information is to be disregarded (forgotten).

$$f_i^{(t)} = \sigma\left(\sum_i w_{k,i}^{fh} h_i^{(t-1)} + \sum_i w_{k,i}^{fx} x_i^{(t)} + b_i^f\right) \tag{3.10}$$

The next operation of the LSTM cell in Figure 3.5 is the input gate, symbolised by vector $\boldsymbol{i_t}$ and also the hyperbolic tangent *tanh* activation, which determines how the current cell state vector $\boldsymbol{C_t}$ is to be updated regarding the new input values from $\boldsymbol{x_t}$ and the previous cell output $\boldsymbol{h_{t-1}}$. The input gate $\boldsymbol{i_t}$ is defined by Eq. (3.11) as:

$$\boldsymbol{i_t} = \sigma(\boldsymbol{W_{ih}} \cdot \boldsymbol{h_{t-1}} + \boldsymbol{W_{ix}} \cdot \boldsymbol{x_t} + \boldsymbol{b_i}) \tag{3.11}$$

The parameter matrices $\boldsymbol{W_{ih}}$ and $\boldsymbol{W_{ix}}$ of Eq. (3.11) represent the recurrent and input vector weights, assigned to the $\boldsymbol{x_t}$ and $\boldsymbol{h_{t-1}}$ vectors, respectively. Once again, the input gate bias parameter vectors are represented by $\boldsymbol{b_i}$.

A vector of new candidate values is created by the *tanh* layer as defined by Eq. (3.12), where $\boldsymbol{W_{cx}}$ represents the input value weight matrix, with the hidden layer parameter weight matrix once again represented by $\boldsymbol{W_{ch}}$.

$$\tilde{\boldsymbol{C}}_{\boldsymbol{t}} = tanh(\boldsymbol{W_{ch}} \cdot \boldsymbol{h_{t-1}} + \boldsymbol{W_{cx}} \cdot \boldsymbol{x_t} + \boldsymbol{b_c}) \tag{3.12}$$

With the candidate vector $\tilde{\boldsymbol{C}}_{\boldsymbol{t}}$ values and input-gate vector values $\boldsymbol{i_t}$ defined, the new cell state vector $\boldsymbol{C_t}$ is defined by Eq. (3.13):

$$\boldsymbol{C_t} = \boldsymbol{C_{t-1}} \odot \boldsymbol{f_t} + \boldsymbol{i_t} \odot \tilde{\boldsymbol{C}}_{\boldsymbol{t}} \tag{3.13}$$

Evidently, the new cell state vector $\boldsymbol{C_t}$ consists of two parts. Firstly, the forget gate values $\boldsymbol{f_t}$ are multiplied element-wise by the previous cell state $\boldsymbol{C_{t-1}}$, so that specific values can be completely disregarded ($f_i^{(t)} = 0$) and others can be remembered ($f_i^{(t)} = 1$), with some values partially remembered ($0 < f_i^{(t)} < 1$). Secondly, $\boldsymbol{C_t}$ is updated by the new candidate values according to the input-gate values assigned ($\boldsymbol{i_t} \odot \tilde{\boldsymbol{C}}_{\boldsymbol{t}}$). Thus, the new cell state (cell memory) is partially forgetting the existing memory and adding a new memory [111].

With the LSTM cell state updated accordingly, the final output vector $\boldsymbol{h_t}$ of the LSTM cell is defined with Eq. (3.14). From this, $\boldsymbol{h_t}$ is a function of the output gate values $\boldsymbol{o_t}$, which implements a read function, as defined by Eq. (3.15). These values are multiplied (element-wise multiplication) by the cell memory $\boldsymbol{C_t}$. With $\boldsymbol{C_t}$ pushed through a *tanh* activation

function (pushes values between -1 and +1) the LSTM cell output $\boldsymbol{h_t}$ effectively determines, which memory values to utilise together with the output vector $\boldsymbol{o_t}$.

$$\boldsymbol{h_t} = \boldsymbol{o_t} \odot tanh\left(\boldsymbol{C_t}\right) \tag{3.14}$$

$$\boldsymbol{o_t} = \sigma\left(\boldsymbol{W_o} \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_o}\right) \tag{3.15}$$

It is important to note that the cell state $\boldsymbol{C_t}$ and the LSTM memory cell output vector $\boldsymbol{h_t}$ is not the same thing, since $\boldsymbol{h_t}$ is a filtered version of the cell state [110]. Finally, it should be evident that it is the gating mechanism that allows the LSTM to model long-term dependencies.

The description of the LSTM-RNN model here follows the outstanding work of C. Olah [110], A. Graves [120] and I. Goodfellow *et al.* [108].

3.3.5 Gated Recurrent Unit Recurrent Neural Network

A simpler variant of the LSTM-RNN is the GRU-RNN, which was first introduced in 2014 by H. Cho *et al.* [121]. The well referenced works of J. Chung *et al.* [111], as well as R. Jozefowicz *et al.* [112] have undoubtedly proven the effectiveness of the GRU-RNN, as compared to the LSTM-RNN model for many scenarios. Before the GRU cell operation is presented, it should be noted that the same mathematical notation is used as for the LSTM memory cell description of Section 3.3.4.

The GRU cell is presented by Figure 3.6. Evidently, the GRU memory cell utilises two gating mechanisms that allows it to model long-term dependencies, namely:

- Reset gate $\boldsymbol{r_t}$: Determines the amount of past information to be forgotten.
- Update gate $\boldsymbol{u_t}$: Controls the amount of past information to be maintained.

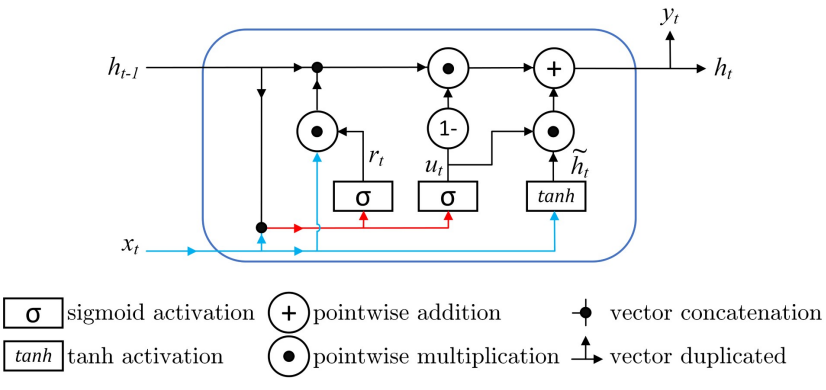


Figure 3.6: GRU cell (*Image redrawn as presented by* C. Olah [110])

Unlike the LSTM cell, the GRU cell does not depend on an internal memory state, which is different from the hidden state. Also, a second non-linearity is not applied to compute the output, as in the case of the *tanh* activation used for the output of the LSTM cell.

From Figure 3.6, Equations (3.16) to (3.19) characterise the inner workings of the GRU memory unit [112]. The reset gate vector $\boldsymbol{r_t}$ is defined by Eq. (3.16), with σ as the logistic

sigmoid activation, together with $\boldsymbol{W_{rx}}$ and $\boldsymbol{W_{rh}}$ as the weight matrices for the input $\boldsymbol{x_t}$ and hidden state $\boldsymbol{h_{t-1}}$ vectors and $\boldsymbol{b_r}$ as the bias parameter vector [121].

$$\boldsymbol{r_t} = \sigma\left(\boldsymbol{W_{rx}}\boldsymbol{x_t} + \boldsymbol{W_{rh}}\boldsymbol{h_{t-1}} + \boldsymbol{b_r}\right) \tag{3.16}$$

The update gate, similar in computation to the reset gate, is presented with Eq. (3.17), with the influence of $\boldsymbol{h_{t-1}}$ and $\boldsymbol{x_t}$ defined by the weight parameter matrices $\boldsymbol{W_{ux}}$ and $\boldsymbol{W_{uh}}$ and bias parameter vector $\boldsymbol{b_u}$ values.

$$\boldsymbol{u_t} = \sigma\left(\boldsymbol{W_{ux}}\boldsymbol{x_t} + \boldsymbol{W_{uh}}\boldsymbol{h_{t-1}} + \boldsymbol{b_u}\right) \tag{3.17}$$

Similar to training the LSTM-RNN model, with BPTT utilised to obtain the gradients, the weight and bias parameter values are optimised during training. Finally, the output vector $\boldsymbol{h_t}$ of the GRU cell is defined by Eq. (3.18), which is clearly dependent on the new hidden state vector $\tilde{\boldsymbol{h}}_{\boldsymbol{t}}$, as formulated by Eq. (3.19) [121] as:

$$\boldsymbol{h_t} = \boldsymbol{u_t} \odot \tilde{\boldsymbol{h}}_{\boldsymbol{t}} + (1 - \boldsymbol{u_t}) \odot \boldsymbol{h_{t-1}} \tag{3.18}$$

$$\tilde{\boldsymbol{h}}_{\boldsymbol{t}} = tanh\left(\boldsymbol{W_{hx}}\boldsymbol{x_t} + \boldsymbol{W_{hh}}\left(\boldsymbol{r_t} \odot \boldsymbol{h_{t-1}}\right) + \boldsymbol{b_h}\right) \tag{3.19}$$

Thus, with the GRU output $\boldsymbol{h_t}$ defined by Eq. (3.18), it should be clear that the update gate vector $\boldsymbol{u_t}$ can either completely copy the previous hidden state $\boldsymbol{h_{t-1}}$, or can completely replace the GRU output with the new hidden state $\tilde{\boldsymbol{h}}_{\boldsymbol{t}}$, or linearly vary between these extremes. The role of the reset gate is to determine if the new hidden state $\tilde{\boldsymbol{h}}_{\boldsymbol{t}}$ should be totally replaced with the current input $\boldsymbol{x_t}$, or if the previous hidden state $\boldsymbol{h_{t-1}}$ should be completely ignored ($\boldsymbol{r_t} = 0$) [121]. Thus, $\boldsymbol{r_t}$ introduces an additional non-linearity regarding the relationship between the future and past states [108, pp.400-401].

Ultimately, the GRU is more streamlined, which is why it has less representational power than the LSTM, but is also computationally less expensive to run [70, p.215]. This trade-off between computational expense and representational power is an almost constant design consideration for the process of developing the forecast models. Finally, the implementation of the GRU-RNN for this research is illustrated with Figure 3.7. With the input sequence provided to the GRU-RNN model, the final output of the stacked GRU layers are passed on to a fully connected output layer, with a ReLU activation for each of the output time-steps.

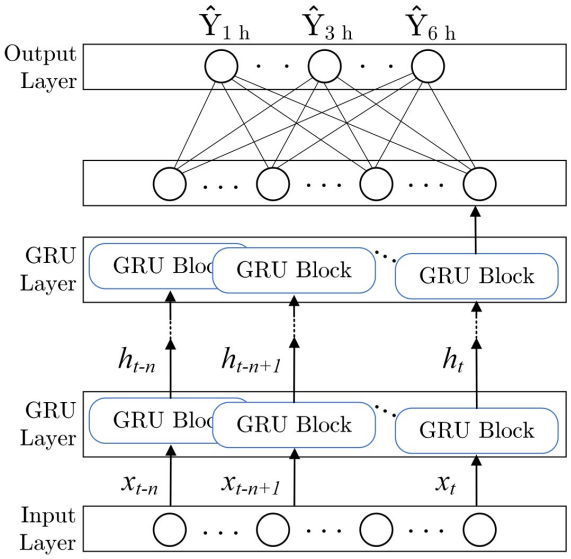


Figure 3.7: GRU-RNN model implementation.

To conclude on the difference of the RNN, LSTM-RNN and GRU-RNN models, the traditional RNN always replaces the content of a unit with a new value determined from the previous hidden state and the current input. However, the LSTM-RNN and GRU-RNN model units always keep the existing content of the hidden units and simply add the new content [111].

3.4 Input feature selection

Model development naturally starts with the selection of input features. With the weather and power output data sets prepared as discussed in Chapter 2, together with the additional input features, the final stage of data processing is performed in parallel with model development. As evident from the discussed literature in Section 1.2.3, PV-system power has a strong dependence on various meteorological factors. There is unfortunately no universal set of input features, which can be generically applied for the development of PV-forecasting models. The reason being that PV systems located in areas, each with a unique set of environmental attributes. For example, PV systems located in areas with frequent snowfall, utilise snowfall as an input feature. On the other hand, wind speed and direction as input features have been proven vital in some scenarios [52], whereas other scenarios have indicated no significant performance increase due to wind data [84].

Simply using all of the available variables as input is not an optimal strategy, since there is a delicate balance between the set of input features and the increased model complexity and computational expense [15]. Only highly correlated data should be considered for anticipated PV-system power output [9]. To identify features that are highly correlated to PV-system power output, a logical first step is to determine the correlation coefficients. Essentially, such an analysis measures the strength of the relationship between two variables. The correlation coefficient can range between $+1$ and -1, with ± 1 as the strongest possible degree of association and zero indicating that variables are unrelated [114, pp. 55-56].

A standard method to identify these correlations is the use of the Pearson correlation, which has proven to be a popular choice in PV-forecasting literature [72, 74, 75]. Essentially, this analysis measures the strength of the linear relationship between two variables [122, p. 298]. However, obtaining the correlations with a Pearson analysis, as executed in Appendix A.5.1,

it is assumed that the variables compared should have a Normal (Gaussian) distribution [122, p. 298]. To avoid any potentially incorrect assumptions regarding the data distributions of the available variables, a rank correlation method can be used.

An example of a rank correlation method is the Spearman's rank-correlation, presented in Appendix A.5.2. Unlike the Pearson's correlation, which strictly measures the strength of linear relationship, it measures the monotonic relationship, where an increase in one variable corresponds with the increase in another [123, 124]. The Spearman's correlation coefficient is also a distribution-free (non-parametric) correlation statistic.

For this research, in an attempt to identify and prioritise input features for optimal model solutions, a full correlation analysis is executed for both methods. This is demonstrated in Appendix A.5. From the discussion presented and the graph of Figure A.40 it is clear that there is an added complexity regarding the selection of historic model input features. For example, the use of humidity data measured 6 h ago as input feature, might prove to provide more value than the data captured for the previous 1 h or 12 h time-steps.

However, input feature selection is dependent on more than just correlations. The additional factors to consider regarding the selection of input features can be itemised as follows:

- *Input feature correlation*: How well the input feature behaviour correlates with power output dynamics.
- *Historic data requirement*: The number of historic input feature time-steps (lagged observations) required for a feature to deliver useful information in the anticipation of the future state of PV-system power output.
- *Input feature synergy*: Scenarios where the combined value of certain features is greater than the individual value of the features. For example, wind speed & direction vs. wind direction only.
- *Model receptiveness*: The ability of a model to handle the added complexity of more input features. As shown in Sections 3.2 and 3.3, some models are theoretically limited in the ability to absorb and use input features to deliver forecasts. For example, FFNN models that receive a vector of input features and RNNs, which unroll the 3D input feature data set, thus enabling RNN-based models with a higher input capacity for input features.
- *Forecast horison*: The forecast horison not only determines the historic input data time-steps, but also the true relevance of input features. For example, as was shown with the literature review, forecasts made 1 h ahead typically do not require any other data than the most recent power output, whereas forecasts delivered 3 h ahead have shown to benefit from historic power and temperature data.

Therefore, although useful, a correlation analysis alone as for example applied by Y. Wang *et al.* [74], cannot adequately capture the complexity regarding the influence of individual input features on the forecasting accuracy. For these reasons, the best strategy to define the optimal input feature set is the use of an iterative process of feature selection and sliding-window size (lagged observations) selection.

3.4.1 Sliding window

As mentioned, although the correlation analysis might have given some indication regarding the strength of association between power output and individual input features, there is

no certainty regarding which input features to utilise. Further, there is also no method of analysis, which defines how many historic data points of each input feature is required to deliver the most accurate forecasting models.

With the use of a standard sliding-window approach (also described as look-back window), previous time-steps of measured data is grouped as a single subset of data, which serves as input for the models to be trained on. The use of a historic sliding-window size as applied for this research, given a set of input features, is visually demonstrated with Figure 3.8. The sliding window, referred to as the HISIMI + xh (Historic Similarity and xh), serves the unique purpose to provide a regular sliding-window subset of data, combined with the historic power output data measured for the exact forecast time window power output (5 h window) size, as delivered 24 h ago.

From an iterative exploration of various input features and sliding-window size combinations the optimal input data set, which yields the highest forecasting accuracy can be determined. However, since there is no generic method to predetermine the input feature and sliding-widow combinations, this data is uniquely prepared for each iteration of model development.

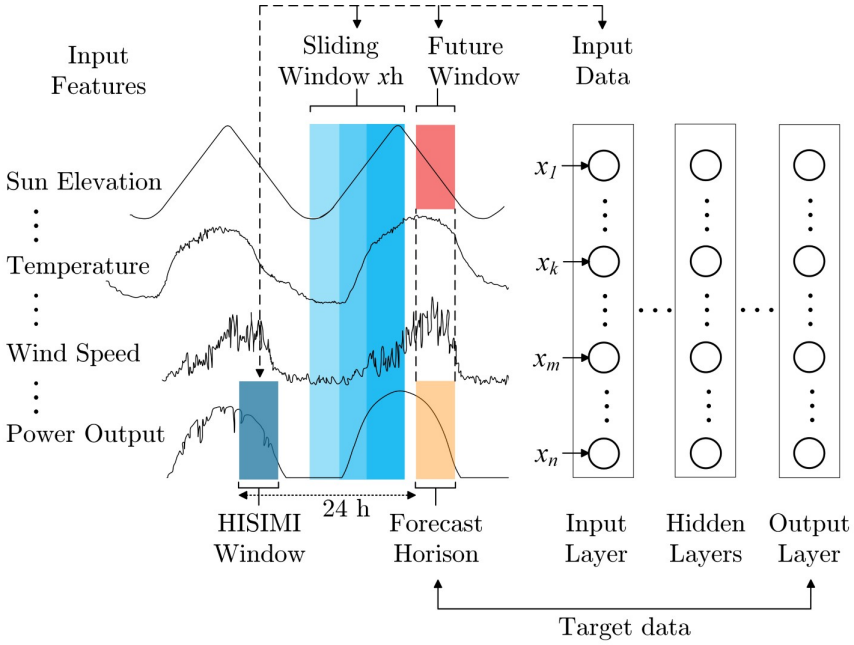


Figure 3.8: Example of multivariate sliding window approach applied for data preparation.

As mentioned, the input features consist of two groups, representative of historic (power and weather data) and future data (sun position, time-related data). For the future-based data, a fixed sliding-window size of 5 h is adopted, equivalent to the forecast horison, since all of the data is relevant to the forecast horison. Regarding the historic input data, there are various selections available for sliding-window sizes. However, for this research the following historic sliding-window combinations are used:

- 1, 2, 3, 6, 24 h, HISIMI + xh

These time segments are chosen, partly due to the information obtained from the Spearman correlation analysis, but also since it represents a good sample set of overall intra-day historic values. From empirical analysis these sliding-window sizes are found to represent a wide

enough range of historic data, adequate for intra-day forecasting. The research of D. Lee *et al.* [71] is also in agreement with these sliding-window sizes for short-term forecast models.

Ultimately, the HISIMI + xh sliding window combination exploits the historic similarity of power output, using it together with the most recently measured data, which provides additional information about the immediate environment conditions. For example, should the most accurate model for a given set of input features (power, GHI, temperature) result from a 3 h sliding window, then the data segment measured 24 h ago regarding historic power output (HISIMI window) is used together with the 3 h sliding window. Mentioned in Section 3.4.1, the only input data considered for the HISIMI sliding window is power output.

As reminder, the available input data is presented in Table 3.2. With the research aim to compare macro-level and aggregated inverter-level forecasts, two power output data sets are used, as demonstrated in Figure 3.9. The inverter-level models are each dedicated to and trained with power data from the corresponding inverter, whereas the macro-level models are trained on the global PV-system power data, representative of total power output. Both processes of macro-level and inverter-level model development are executed with the same weather input features (T, GHI, WS, etc.). Further important is that both the macro-level and inverter-level models are trained on the same amount of time-correlated data.

Table 3.2: Final set of available input features.

Input feature	Description	Symbol	Unit
	Historic information		
Power	Global power output used as historic input data for macro-level models. Historic power output of individual inverters serves as input for each inverter-level model.	P	[W]
Irradiance	Global horisontal irradiance measured on site.	GHI	[W/m^2]
Wind speed	Wind speed measurement.	WS	[m/s]
Wind direction	Provided as one-hot encoded vector of four wind directions.	WD	[°]
Relative Humidity	Measure of water vapour present in the air relative to the amount required for saturation at the same temperature.	RH	[%]
Absolute Air Pressure	Atmospheric air pressure exerted relative to a vacuum.	AP	[hPa]
	Future information		
Solar azimuth	Angular position of the sun, measured as East or West of North as reference line.	β	[°]
Solar altitude	Angular measure of the geometric centre of the sun relative to PV system horison.	Φ_S	[°]
Month	One-hot encoded vector indicating the month of the year.	–	–

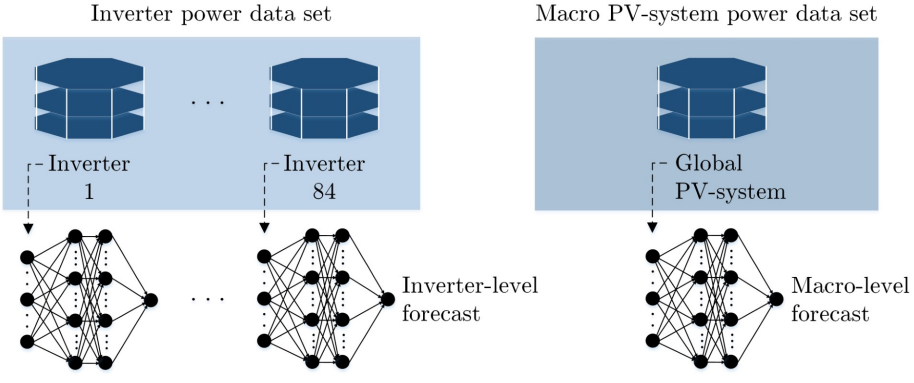


Figure 3.9: Distinction between the inverter and macro PV-system data sets.

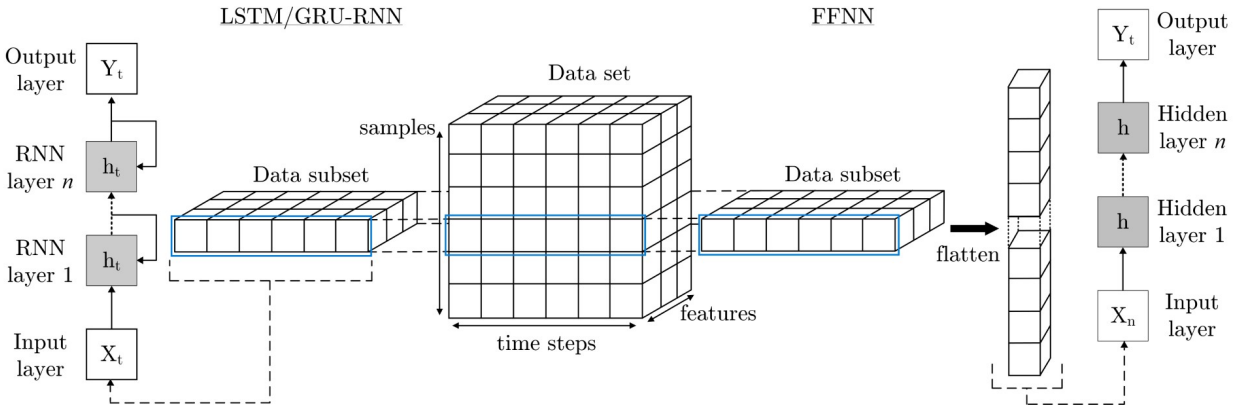


Figure 3.10: Illustration of how data set samples are provided to GRU-RNN, LSTM-RNN (left) and FFNN (right) models as input data.

With the data ready, it is prepared as a 3D input data set for the models, consisting of the number of samples, number of time-steps and the number input features (sample × time-steps × features). From this primary data set, a sampled subset of data is selected, which then serves as input for the models. For the FFNN implementation the input data set is flattened, as demonstrated with Figure 3.10. However, regarding the GRU-RNN and LSTM-RNN model implementations, the input data is maintained as a 3D subset as illustrated with Figure 3.10.

Finally, given the inherent seasonality of the data set, the order of the data has significance. For example, the segment of GHI and temperature data within the square box in Figure 3.11 has a clear upwards trend. Should only this data segment be used for model training, an extrapolated trend (dashed lines) can be anticipated by the model output, resulting in sequence bias. To avoid this, a good strategy is to shuffle the training batch samples, with the time-steps and feature sequences still maintained. This allows the model to focus on what is truly important, which is the relationship between the input features and the target output. However, given the training set used for this research, no noteworthy difference was seen between shuffling or not shuffling the training data. It was concluded that the training set of two years allowed the models to be exposed to full cycles of seasonality. Therefore, leaving little chance for exposing the models to a definite trend or sequence bias as in the Figure 3.11 example.

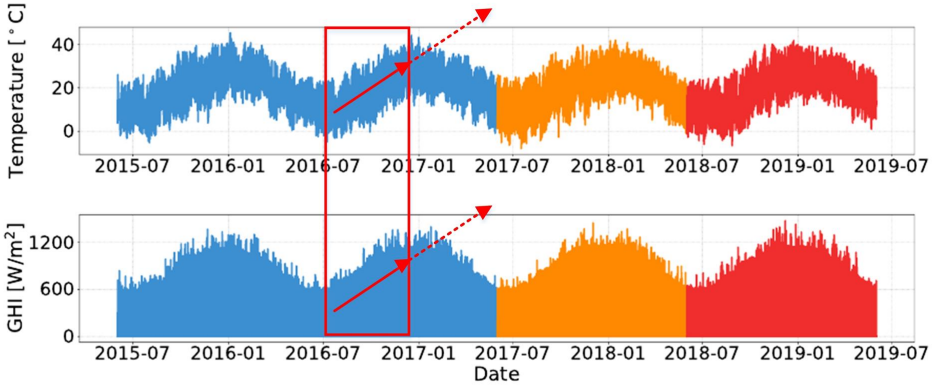


Figure 3.11: Illustration of a data segment with a clear trend.

3.4.2 Feature scaling

Feature scaling is utilised for the training of models due to the various numerical ranges of the input features, which results in faster convergence of the training process. Two common data-science approaches utilised for feature scaling are *min-max scaling* and *standardization* [114, p.65]. With both feature scaling techniques applied, it is determined that models performed somewhat better with data normalised by means of the min-max scaling technique. Therefore, the input features (except for the one-hot encoded features), are normalised with this scaler, defined by Eq. (3.20), so that data set x is now confined as $\tilde{x}\epsilon[0,1]$.

$$\tilde{x} = \frac{x - min(x)}{max(x) - min(x)} \tag{3.20}$$

Finally, the variables of the scaler, which are $min(x)$ and $max(x)$, are determined from the training data only and are determined independent of the validation and testing data sets. This is done, to keep the test data set completely unknown to the developed models. Thus, the normalisation values ($min(x)$ and $max(x)$) determined from the training set are used for scaling the validation and test set data points.

3.5 Model training and fine-tuning

3.5.1 Optimiser, learning rate and mini-batch size

The model parameters (weights, bias values) are fine-tuned with the use of the back-propagation algorithm and a Gradient Descent (GD) optimiser. The Adam optimisation algorithm is used, originally introduced by D. Kingma and J. Ba [125], which was found to perform better than the regular Stochastic Gradient Descent (SGD) algorithm. The exponential decay rates for the 1st and 2nd moment estimates are kept constant at 0.9 and 0.999, respectively.

Specifically, the Adam optimiser has been well cited within the ML-research community to be fairly robust regarding the choice of learning rate (LR) and hyperparameters [108, p.302]. From experimentation, several values for the LR ranging between $1e^{-4}$, $1e^{-3}$, $1e^{-2}$, $1e^{-1}$ were selected for model development, with a LR of $1e^{-}4$ proven to be a good selection for the FFNN, GRU-RNN and LSTM-RNN models. Therefore, all models are fine-tuned with the use of the Adam optimiser, with a LR of $1e^{-4}$.

Regarding the mini-batch (MB) size, the recent work of D. Masters and C. Luschi [126] proved the effectiveness of smaller MB sizes, with values ranging between 2 and 32 recommended. As mentioned in their research, the gained advantage of increased GPU processing power from utilising large batch sizes for SGD-based training, is offset by the ability of smaller batch sizes to allow for better model generalisation, as well as ensuring a smaller memory footprint [126]. Interestingly, it is also shown that the use of smaller batch sizes (opposed to thousands of MB samples) allows for a better range of learning rates.

For the development of the PV-forecasting models, empirical analysis revealed that MB sizes of 32 and 64 perform the best, combined with the Adam optimiser and the chosen LR of $1e^{-4}$. However, as further discussed in Section 3.6.2.3, other mini-batch sizes are also considered for the model development.

3.5.2 Activation function

As thoroughly discussed in Section 3.2.2, the ReLU activation function is used, since it significantly improves the convergence speed of training algorithms [101]. Also, the ReLU activation ensures that power output forecasts are never below zero, since negative power would suggest that power is being consumed by the PV system. Although this scenario is probable (with inverters and other equipment consuming power during night time), the aim is to predict PV-system power output, as delivered to the electrical grid.

Regarding the implementation, the ReLU activation is used in each hidden layer and the output layer of the FFNN models developed. For the LSTM-RNN and GRU-RNN model implementations, the final output layer is a fully connected layer, with ReLU as activation function.

3.5.3 Early-stopping and regularisation

With the training and validation data sets established, model parameters (weights and bias) are optimised until the MSE cost function, discussed in Section 3.2.3, is minimised. To avoid overfitting of the training data and ensure that models generalise well on the unseen test data, an early-stopping strategy is applied. After each epoch of training, model accuracy is evaluated for the validation set. A patience (waiting) strategy is also applied, so that when forecasting accuracy on the validation set stops improving consecutively for n epochs, training is stopped. With this approach, the decision regarding the number of epochs to train on is set as an arbitrary high number of 3000, with a patience value of 20 epochs found to be more than adequate.

Regularisation by means of a drop-out strategy is also applied, with various drop-out values ranging between 0.1 - 0.5 investigated. However, since no performance increase resulted from drop-out regularisation, it is not used for model development.

3.5.4 Data reproducibility

For this research, data reproducibility is crucial for two reasons. Firstly, as a research project it serves the purpose of exactly reproducing the results obtained, given the same data. Secondly, by controlling the pseudo random number generators, it is easy to determine whether improvements in model accuracies actually result from a different selection of hyperparameters, or if it is simply due to another set of model parameter initialisations by the kernel initialiser (used Glorot normal initializer, also known as Xavier normal initializer).

3.6 Hyperparameter optimisation

Apart from the hyperparameter selection of LR and MB size, hyperparameter tuning regarding the number of hidden units (HUs) and hidden layers (HLs) are what dictates the model performance to a large extent. During experimentation, it was consistently found that changes in the architecture lead to the biggest performance increases, further supplemented with a good LR and MB size. Unfortunately, similar to MB size and LR selection, there is no set theory regarding the decision of how many HLs and HUs to use, making the selection of these hyperparameters difficult empirical work. Therefore, a systematic approach is required to explore different configurations.

3.6.1 The issue of claimed model superiority

It was concluded from the literature review presented in Section 1.2, that there exists a great disparity within the PV-forecasting research field, with almost all scholars claiming the superiority of their proposed models. It was mentioned that these claims of superiority are often founded on completely different scenarios, with four factors listed, which highlight the root cause of what can be considered unfair claims. The two biggest contributing factors towards this research disparity are identified as: 1) The use of incomplete test data sets and 2) Biased model optimisation efforts.

Regarding the second factor of biased model optimisation, various researchers have compared their highly optimised models, such as a novel hybrid or stand-alone ML-based solution, to models that did not receive nearly the same amount of optimisation effort. This is obvious from most published research, where researchers mostly only specify the training process applied towards parameter optimisation. However, in almost all cases authors completely fail to specify the process of hyperparameter optimisation for the models applied, with specific reference to the number of HUs, HLs, MB size and LR.

The reason why researchers have been able to publish, irrespective of these shortcomings, is attributed to hyperparameter selection being a highly iterative process, often executed based on intuitive experimentation. This is in sharp contrast to adopting a reproducible and systematic approach of applied hyperparameter optimisation. Although experienced researchers might have a good intuition of guiding models towards accurate solutions, without a systematic process to formally account for solutions considered and effort applied, biased solutions are inevitable.

Also, the lack of a formal hyperparameter optimisation framework makes it incredibly difficult for other forecasting practitioners to gauge and identify possible ML-based solutions from published works, which are applicable to their own scenario. Therefore, total transparency and a systematic and formal account of model optimisation is an objective of this research, so that trustworthy and accurate claims of model forecasting ability can be made.

3.6.2 PV-system macro-level forecast model development

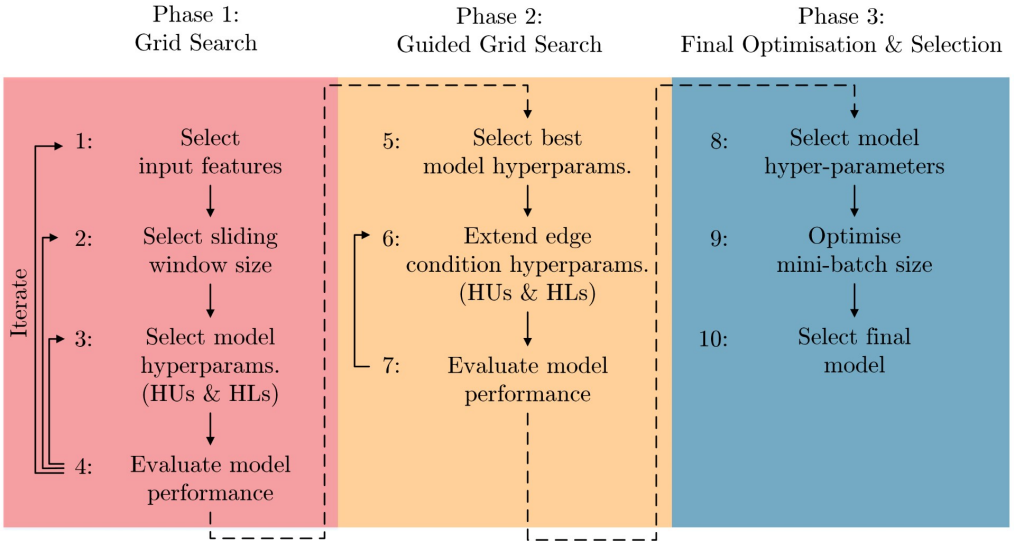


Figure 3.12: Macro-level forecast model development framework.

This goal of delivering unbiased model comparisons is achieved with the use of a model development framework, which systematically guides the applied effort towards finding optimal solutions. As illustrated with Figure 3.12, the framework consists of three phases, which ensure that all models have been awarded an unbiased opportunity to deliver forecasts.

3.6.2.1 Phase 1: Extensive grid-search

Firstly, a combination of the available input data features (historic weather and power data) and sliding-window sizes (1 h, 2 h, 3 h, 6 h, 24 h, HISMI + xh) is selected. The combinations considered are displayed in Table 3.3. In total, there are seven unique historic input features, which provide weather related and power output information, together with three future-based features, which provide information regarding future sun position and the month of the year. Six historic sliding windows are also applied during training, as motivated for in Section 3.4.1, with the HISIMI + xh (Historic Similarity and x h) sliding window consisting of the information obtained from a sliding-window size of x h, combined with the same forecast time window power output, as delivered 24 h ago.

Table 3.3: Input feature and sliding window combinations.

Feature type	**Input features**
Historic	P, GHI, T, WS, WD, RH, AP
Future	β, Φ_S, month
Window type	**Window size [h]**
Historic-sliding	1, 2, 3, 6, 24, HISIMI + xh
Future	5 (1 - 6 h ahead)

With the sliding-window and input feature combinations selected, step three of Phase 1 proceeds towards an extensive hyperparameter grid-search [127], which iterates over various HL, HU and MB size combinations. This selection of input feature, sliding window and model hyperparameters are put into context, with Figure 3.13 conveniently displayed again. With an infinite number of available hyperparameter combinations, as demonstrated with Figure 3.14, the grid-search domain must initially be confined to a selected range of values for each hyperparameter. This initial grid-search domain is to be empirically determined from sampled results obtained for various input features, sliding window and hyperparameter (HUs, HLs, MB size) combinations. With this spot-checking process completed, an initial grid-search solution space should be defined.

With the initial grid-search domain established, a standard approach towards training models and identifying the best HU, HL and MB size combination, is by means of a Random-search approach, as presented by the extensively referenced work of J. Bergstra and J. Bengio [127]. Although Random search is ideal for scenarios where time and resources are limited, it does not allow for a full spectrum analysis of all possible HL, HU and MB size solutions, which is why for this research an exhaustive grid-search approach is applied. This ensures that all models are given an equal amount of targeted effort to obtain the best possible solutions, as executed during steps three and four of Phase-1 development.

3.6.2.2 Phase 2: PV-system model

With solutions obtained from Phase-1, Phase-2 of model development expands the grid-search domain for models, which delivered a high forecasting accuracy specifically for hyper-

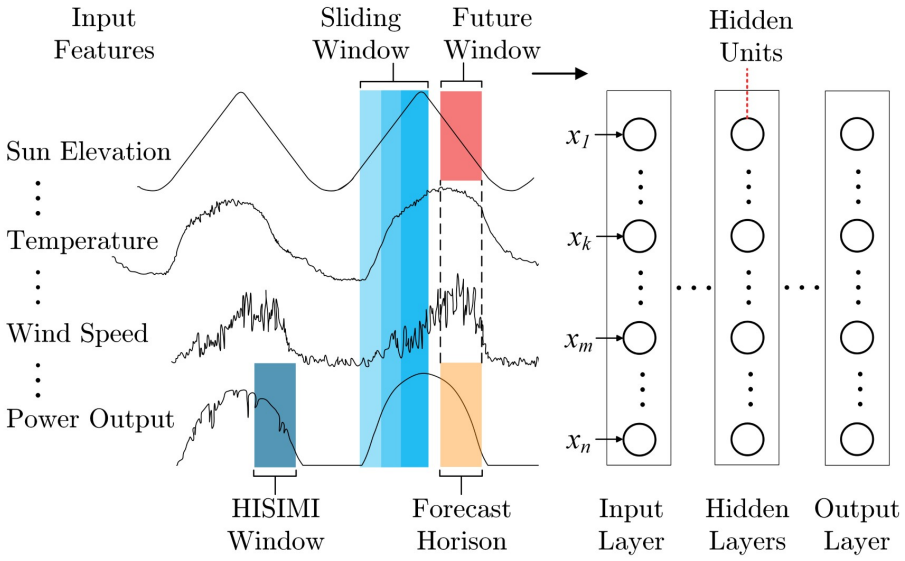


Figure 3.13: Illustration of the relationship between the sliding windows, input features, hidden layers and hidden units.

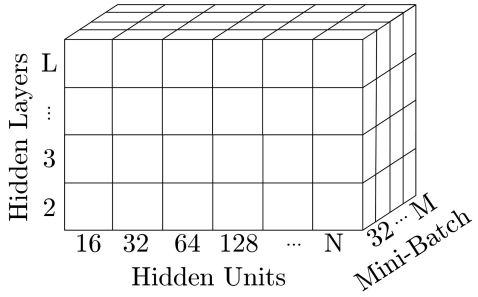


Figure 3.14: Example of a hyperparameter grid-search domain.

parameters on the edge of the Phase-1 grid-search domain. This guided grid-search approach (also referred to as Coordinated descent) is demonstrated in Figure 3.15. For example, an edge-case would be a solution obtained for HU=128 and HL=2, as demonstrated by the initial grid-search domain of Grid A. Therefore, the HU solution domain is extended, as seen in Grid B. With the new best solution obtained for HU=256 and HL=3 in Grid C, both the HU and HL size are edge cases. The HU size is firstly extended, as seen in Grid D. Notice, that when a hyperparameter (HU in case of Grid D) is extended, one smaller and one larger size of the other hyperparameter (HL in the case) is also explored. With no improved results from Grid E and HL=3 still an edge case, the HL size in turn is extended to HL=4, for HU=128, 256 and 512 (but 512 will be ignored, since it has already been explored) as seen in Grid F. This process is repeated until the accuracy saturates in all directions. For this example, this occurs for the solution space of Grid I. With no further improvement, the solution accuracy is saturated and Phase-2 of development is completed. Importantly, this process of a guided grid-search requires that only one variable is changed at each iteration. For example, simultaneously changing both the HUs and HLs will not give a definitive direction of improvement.

Compared to a conventional Random search [127] and unstructured intuitive (organic) approach towards hyperparameter optimisation, the guided grid-search evidently ensures a formal and unbiased process of model development.

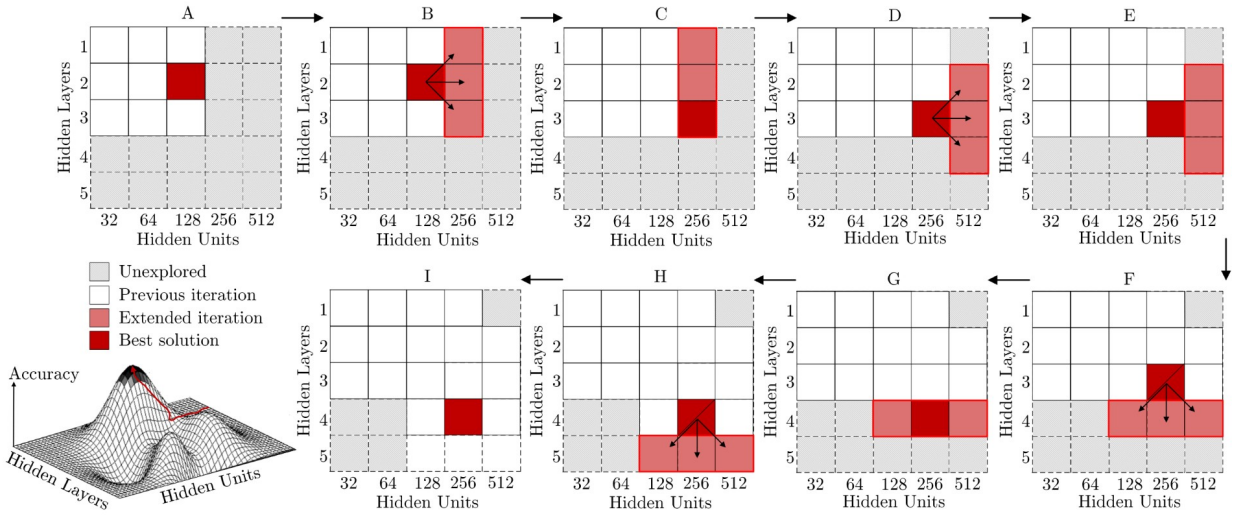


Figure 3.15: Illustration of Phase-2 guided grid-search process, with NRMSE used to determine model accuracy.

3.6.2.3 Phase 3: Final optimisation and selection

Phase-3 of model development is aimed at MB size optimisation. This is regarded as supplementary to model performance, since experimentation has revealed that the largest changes in model accuracies are observed from a change in HUs and HLs. For the development of the forecast models during Phase-1 and Phase-2, the MB size must be maintained for at least two MB sizes, so that a good general solution set is obtained. From experimentation, the MB sizes of 32 and 64 are found to consistently deliver good results. Another reason for the selection of 32 and 64 as initial MB size is that these smaller batch sizes, combined with smaller LRs, allow for regularising effects, which further allow solutions to generalise well [128]. However, to ensure that further improvement in model accuracy is not disregarded, additional model solutions for various other MB sizes are explored. For this research, MB sizes of 4, 8, 16, 128 and 256 are also selected, to ensure that the best solutions are obtained for the given number of HUs and HLs, as identified during Phase-1 and Phase-2 of hyperparameter optimisation.

3.6.3 Inverter-level model development

An unbiased process of model development is required to deliver a fair comparison of the macro-level and aggregated inverter-level forecasting solutions. This is to be achieved by replicating the PV-system model development framework, as defined in Section 3.6.2, for the development of the inverter models.

However, the prerequisite of training 84 individual inverter models with a full-stack grid-search approach, applied to each model, will require a significant amount of time and resources. With the intent of this research to deliver results, which have value regarding a real-world application, it is necessary to consider solutions, which are practically feasible regarding scalability and reproducibility.

3.6.3.1 Inverter-clustering technique

Another solution to reduce training time, but still ensure the optimal hyperparameter (HUs, HLs, MB size) selection for each inverter model, is to divide the PV-system inverters into clusters (also referred to as segments). This is conceptually illustrated with Figure 3.16.

With the inverter clusters defined, only one representative inverter per cluster has to be trained with a full-stack grid-search approach, as executed for the PV-system macro-level models. The hyperparameter selection, which delivers the most accurate forecasts for the representative inverter, is anticipated to be characteristic of the preferred hyperparameter selection for all of the other inverters within the particular cluster.

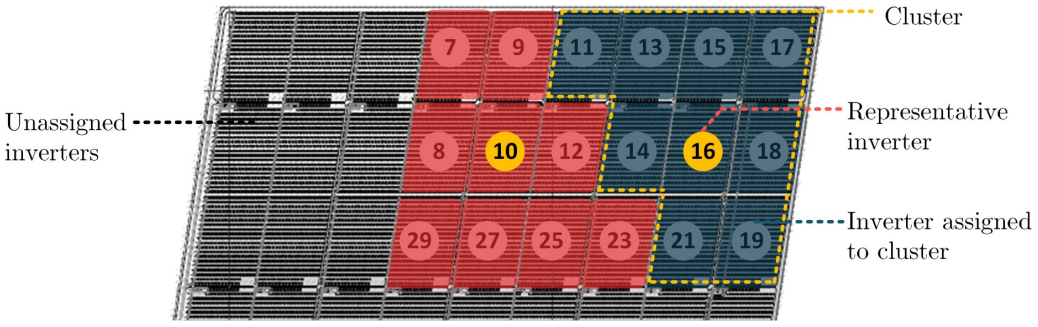


Figure 3.16: Inverter assignment to clusters.

With the power output of each inverter available, the allocation of inverters to each cluster should be based on power output similarity, as defined by two sets of criteria:

A. *High temporal correlation*: Real-time PV power output signal dynamics are different for various locations within the PV-system. This is attributed to physical PV-system size, which enhances environmental factor differences. In particular, partial cloud movement will for example impact the very short-term (minutes) spatial heterogeneity between inverter power output values. Another example would be the intra-hour and intra-day effects of wind circulation or non-uniform inter-row shading on the inverter power output time-series dynamics.

B. *Similar power output capacity*: Inverters allocated to a cluster should demonstrate a similar power output capacity. Power output signal amplitude further amplifies or reduces signal dynamics (frequency, rapid response), which might necessitate a different set of hyperparameters to be effectively modelled. Specifically, the focus here is on semi-permanent (weeks, months) and intra-week differences in power output capacity and therefore dynamics. Factors which will influence power output capacity are for example inter-system dust distributions, which represent semi-permanent (weeks to months) non-homogeneity. Another influential factor would be the overall effects of dominant (intra-week, intra-month spatial heterogeneity) wind movements within the system, which will further influence overall inverter power output capacity.

With the criteria defined, a metric is required to establish the homogeneity between the power output time-series signal dynamics of each inverter. For this purpose, the Euclidean distance (ED) is selected. The ED as similarity metric has also been proven to perform better than most of the well known time-series clustering distance measures [129]. Another popular metric for identifying time-series similarity is Dynamic Time Warping (DTW) [130]. However, DTW is ideal for time-series data sets that are not of equal length, which also contains out-of-phase similarities [130]. Since both out-of-phase and amplitude differences must be penalised, the ED is ideal to be used as the metric of similarity between the various time-series data sets. Equation (3.21) defines the ED between two time-series data sets, each represented as a vector [131]:

$$ED\left(p_m, p_n\right) = \sqrt{\sum_{k=0}^{N} \left(p_m(k) - p_n(k)\right)^2} \tag{3.21}$$

From Eq.(3.21), $p_n(k)$ and $p_m(k)$ represent the k'th data point for time-series power data set n and m, respectively. The maximum number of time-series data points is presented by variable N, which implies that both time-series data sets m and n must have the same number of data points.

To be exact, for the time-series clustering of the various inverters, a Whole-clustering approach [132] is adopted, where the entire length of each inverter time-series is considered for the ED as similarity metric. The similarity of each inverter is obtained, relative to every other inverter. Thus, with a total of 84 inverters, an ED similarity matrix is obtained with a dimension of 84×84. These individual measures of similarity are presented with Figure 3.17, where blue signifies a high similarity and red a weak similarity.

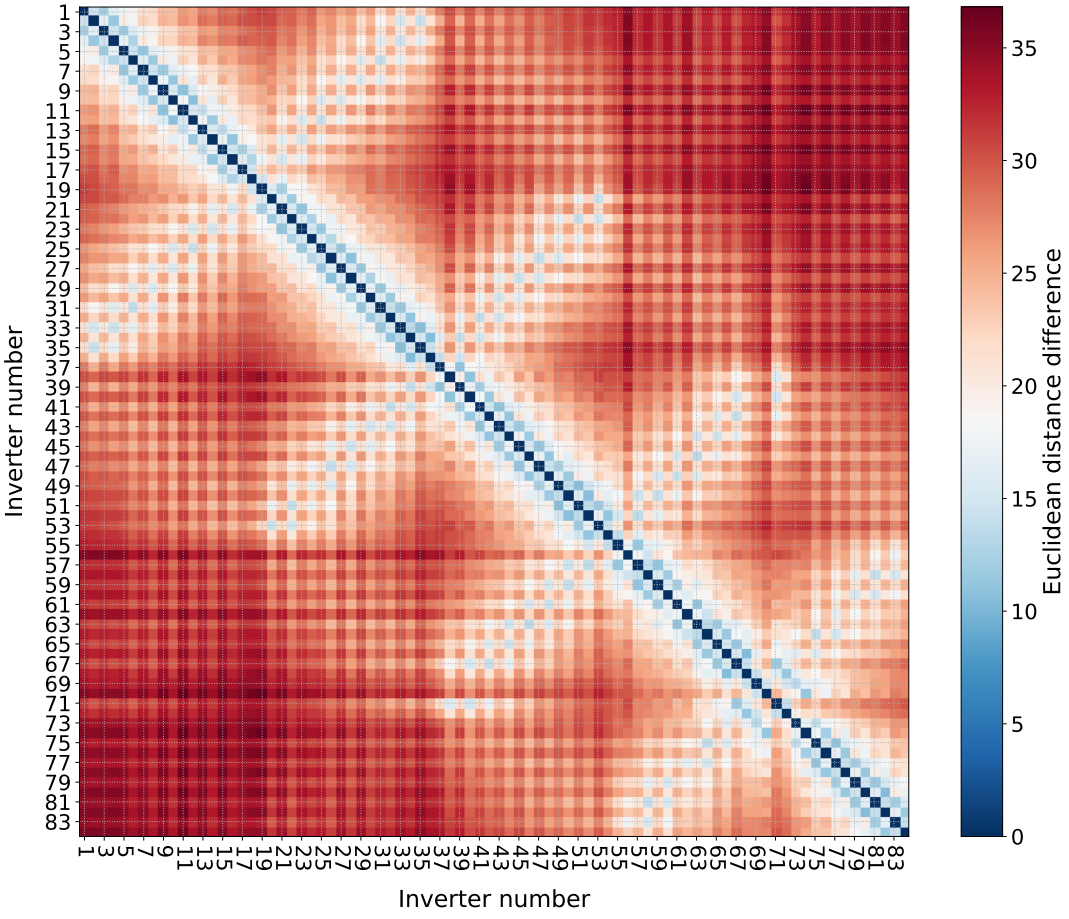


Figure 3.17: Visual representation of Euclidean distance similarity as determined for each of the 84 inverter power output time-series data sets.

Immediately evident from Figure 3.17 is that each inverter has the highest correlation with its own power output data. Also, considering the PV-system layout as presented in Figure 3.18, it is further evident that inverters physically located closest to one another, also have the highest time-series power output similarity.

With the multi-dimensional ED similarity matrix established for each inverter, the inverter clusters are identified with the unsupervised K-means clustering algorithm [133]. A pre-

requisite for using the K-means clustering algorithm is to specify the number of clusters (K). Selecting too many clusters, will result in unnecessary training time and selecting too few clusters might not deliver models capable of sufficiently capturing the various low-level PV-system dynamics.

There are several well documented guidelines of techniques, which help with the decision regarding the selection of K, such as the Elbow method [134], the CH-index [135] and the Gap statistic [136], to name a few. To acquire some indication of what constitutes as an adequate number of clusters, all three of these guidelines are applied, as presented in Appendix G. From these results, K is empirically selected as 10.

With ED as similarity metric and 10 clusters selected for the K-Means clustering algorithm, the Scikit-learn python library is used to perform the clustering. Importantly, the K-Means algorithm is applied with 500 iterations and 20 different centroid seeds to deliver a truly representative solution regarding the inverter allocations to each cluster.

This segmentation of the PV-system inverters into 10 clusters, based on the ED similarity matrix values, is illustrated with Figure 3.18. Evidently, these segments are well distributed within the PV-system. Since there is no exact science regarding the selection of clusters, empirical proof will be provided, to show that this cluster-based hyperparameter selection process is optimal. This is done with a comparison to a scenario where the hyperparameter selections of each inverter are also individually optimised. The results of this extensive analysis, which demonstrates the success of this clustering approach, is presented in Section 4.4.

With the clusters identified, one inverter is selected as the representative inverter for each cluster. The strategy for selecting these representative inverters is simply based on selecting an inverter relatively in the centre of each cluster. In particular, the 10 representative inverters chosen are I04, I10, I16, I39, I47, I53, I67, I59, I74, I81, as indicated by the yellow circles in Figure 3.18.

Therefore, with the number of clusters established and the PV-system model development process (Figure 3.12) as guideline, the inverter-level model development framework is defined as presented with Figure 3.19.

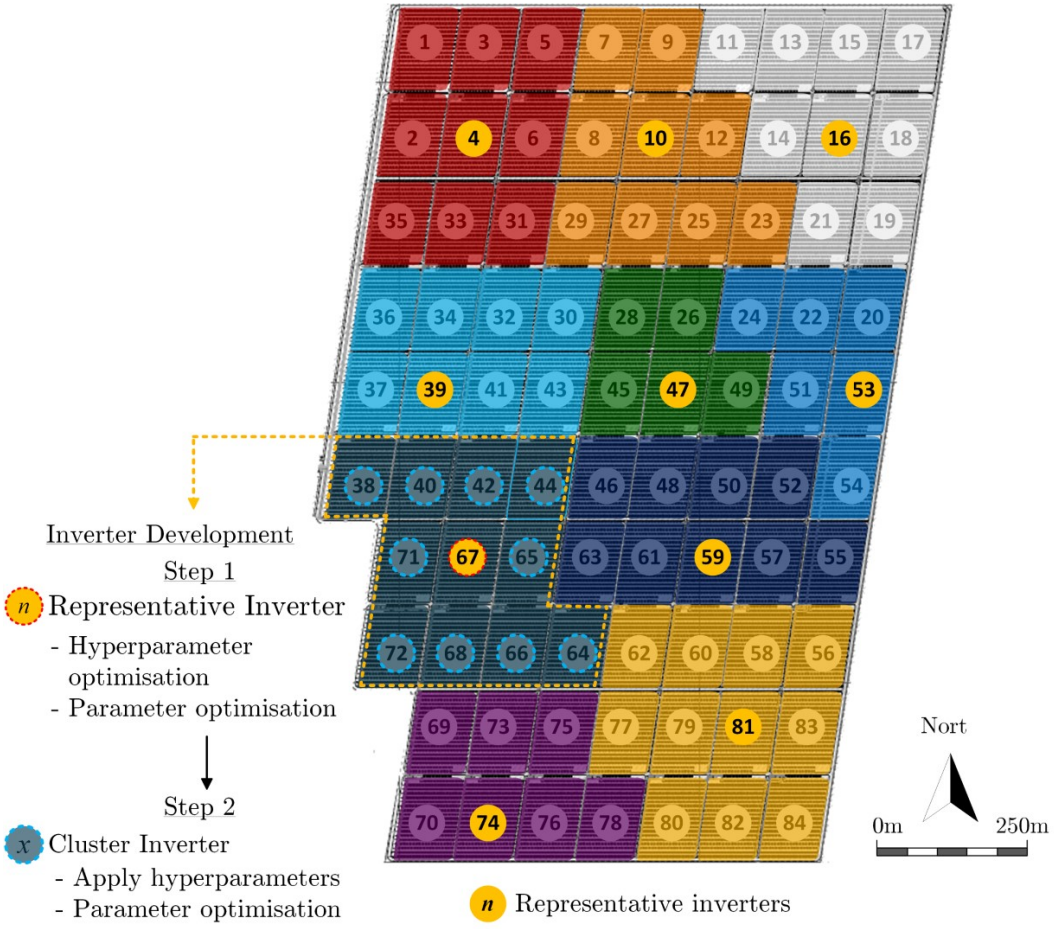


Figure 3.18: Division of inverter blocks into 10 clusters with the systematic optimisation of inverter models for a specific cluster illustrated.

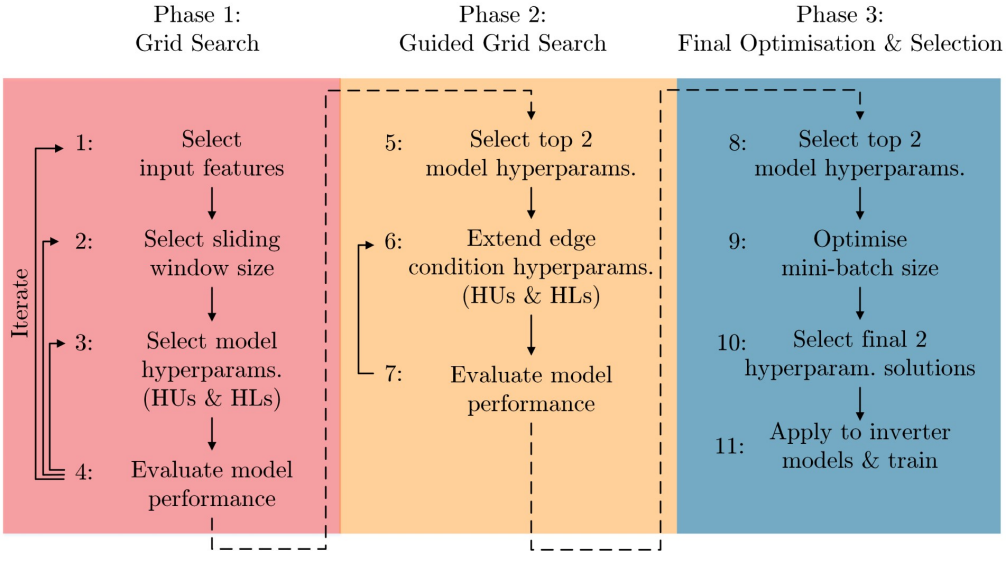


Figure 3.19: Forecast model development framework for inverter models.

In comparison to the macro-level model development process, this framework is identical, except for Phase-2. Instead of only selecting a single top performing hyperparameter selection, the top two hyperparameter combinations are considered for further expansion and

not only the single combination with the highest accuracy. This is done to reduce some of the uncertainty of selecting the optimal hyperparameter set for the inverters assigned to a cluster, based on the preference of the representative inverters. This slight adjustment in the process of inverter-model development does not bias results in any way and still ensures a platform where fair model comparisons are made.

Finally, with the optimal hyperparameter selections obtained for each representative inverter, Step-11 of the model development framework in Figure 3.19 is to be executed, as demonstrated with Step-2 in Figure 3.18. The implementation of these macro-level and inverter-level model development frameworks are further presented in Sections 4.3 and 4.4.

3.6.3.2 Inverter-power-loss emulator

Important to note is that the individual inverter forecast models are trained with the inverter power output as the target data. The macro-level models on the other hand are trained on the macro PV-system power output as target data, which is measured at the point of grid connection. Both the macro-level models and the aggregated inverter-level (84 inverters) model forecast accuracies are evaluated on the macro PV-system power output as target, since this is the true representation of the overall PV-system power output.

Ideally, the sum of the power output for all the inverters should be equal to the measured global PV-system power output. However, inverter-power output has to pass through several transformers and copper cabling. With conduction losses (I^2R losses) and transformer power losses to be accounted for, simply summing together all of the inverter power forecasts, will leave these electrical power losses unaccounted for. Proof of these electrical losses is seen in Figure 3.20. From the close-up bottom graph, it is evident that the summed inverter output is larger than the total PV-system power delivered onto the electrical grid, which demonstrates the power losses.

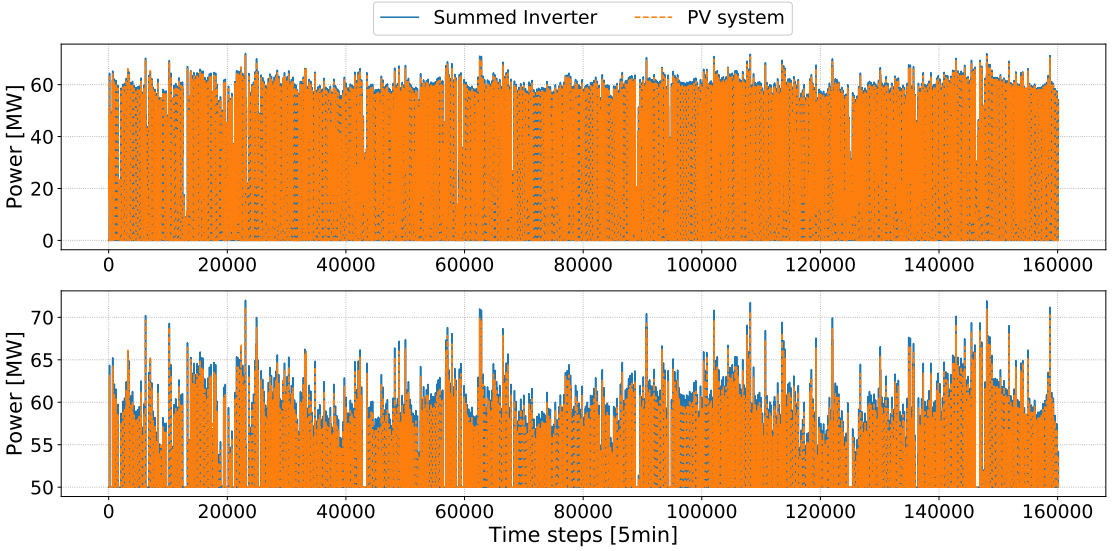


Figure 3.20: Top: Difference between summed inverter power and macro PV-system power output. Bottom: Power output data greater than 50 MW displayed to emphasise difference.

This difference between aggregated inverter-level power and total PV-system power is defined with Eq. (3.22), where P_{loss} represents the power losses, PV_{macro} represents the global PV-system power output and P_{Inv-i} represents the inverter power output of inverter i. Figure 3.21 further demonstrates these power losses, with the relationship of the aggregated

inverter power output (y-axis) plotted against the macro-level PV system power output (x-axis). Evidently, from this graph the difference between these power sources is linear, with a marginal non-linear movement present beyond the 65 MW point of power output. This non-linear movement is attributed to I^2R losses, which are more prevalent at these high levels of power delivery.

$$P_{macro} = \sum_{i=1}^{84} P_{Inv-i} - P_{loss} \tag{3.22}$$

Therefore, the Inverter-Power-Loss (IPL) model emulates P_{loss} as power is transferred from the inverters to the point of grid connection. To train the IPL emulator, historic inverter-power output is summed, which serves as input data for the IPL model. The macro PV-system power output (P_{macro}) serves as target data. With the IPL emulator initially implemented with a Linear regression model, the use of a FFNN model delivered more accurate results. Using the training and validation data sets, the FFNN is developed with a full-stack hyperparameter grid-search strategy, with ReLU as activation function and the Adam optimiser with a LR of $1e^{-4}$. With MSE as loss function the best FFNN architecture is obtained with 4 HLs, 256 HUs and a MB size of 64. This correction of inverter-level forecasts with the IPL emulator model is conceptually illustrated with Figure 3.22, with the actual power output corrections made displayed in Figure 3.23.

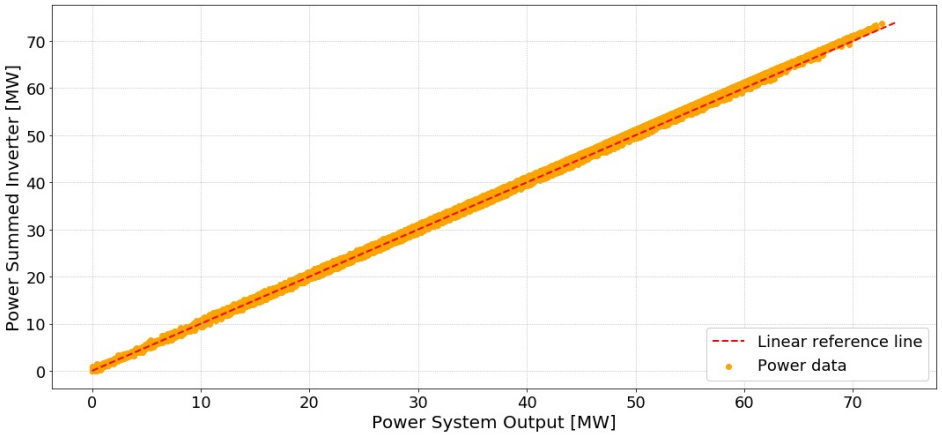


Figure 3.21: Aggregated inverter power output directly compared to measured macro-level PV system power output.

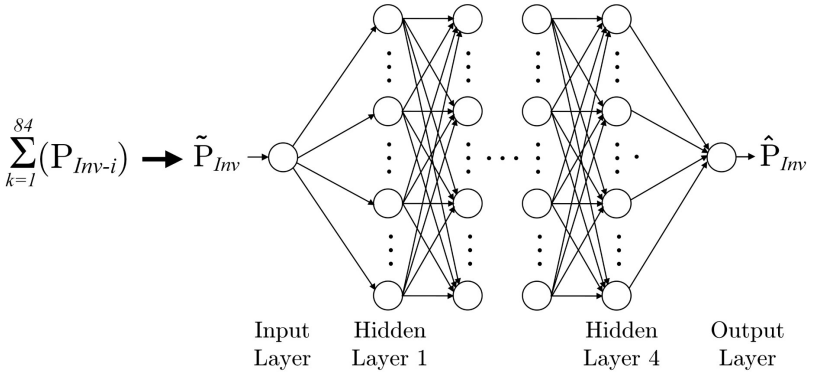


Figure 3.22: Correction of aggregated inverter-level forecasts with IPL model.

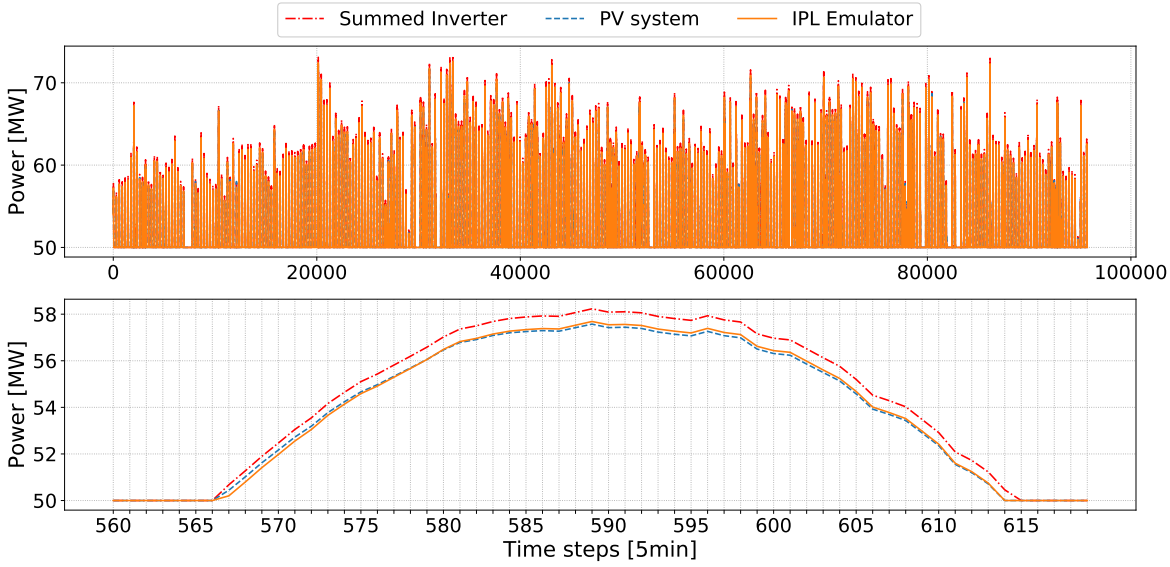


Figure 3.23: Top: Total inverter power output validation set corrected. Bottom: Close-up plot of the aggregated inverter forecasts corrected to account for power losses.

Chapter 4

Results and discussion

4.1 Overview

In this chapter the results of the developed forecasting models are presented. Complete transparency is given regarding the step-by-step execution of the strategic model development framework, as was defined in Chapter 3, which ensures that models are developed with equal effort and that results are unbiased.

Considering the primary focus on macro-level vs. aggregated inverter-level forecasting, each of the DL model accuracies are directly compared in terms of these forecasts. From the results analysis, a final verdict is given regarding the ability of state-of-the-art DL models to capture low-level PV-system power output dynamics. For completeness, direct inter-model comparisons are also made to show how these DL models performed relative to one another.

With the results presented, final remarks are made regarding the real-world day-time-only forecasting accuracies obtained.

4.2 Evaluation metrics

Before any of the results are presented, this section serves as a brief reminder of the evaluation metrics used during the process of model training and selection.

4.2.1 Deterministic

The deterministic evaluation metrics presented in Section 1.2.4 are conveniently repeated here, with the NRMSE and MAPE definitions once again defined by Eqs. (4.1) - (4.2), respectively. As reminder, N is the number of time-series points, i is the i-th element of the predicted power ($\hat{P}_i$) and measured power (P_i) output values, with the installed PV-system capacity ($P_{capacity}$) as denominator.

$$NRMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\hat{P}_i - P_i}{P_{capacity}}\right)^2} \cdot 100\ \% \tag{4.1}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{P}_i - P_i}{P_{capacity}}\right| \cdot 100\% \tag{4.2}$$

As previously explained, each of these metrics serve a specific goal during model development. The NRMSE serves as primary metric during training and for the selection of models, since it aggressively penalises large forecast errors. This ensures that context is maintained of a real-world forecasting scenario, where large errors have more adverse effects (added stress onto electrical grid, financial loss, etc.). Thus, the NRMSE acts as a filter for avoiding models with large forecast deviations.

Once the best individual FFNN, LSTM-RNN and GRU-RNN models have been identified from the solution pool, the final inter-model performance comparisons are evaluated with the MAPE metric. Reason being that the MAPE delivers a good generalised result of typical model performance, which allows for a direct and easy interpretation of model accuracy.

4.2.2 Uncertainty analysis

By quantifying the inherent uncertainty of the deterministic point predictions, more insight is obtained to further assist decision makers [37]. From PV-forecasting literature, the uncertainty associated with the point predictions has been quantified with the use of Prediction Intervals (PIs) and Confidence Intervals (CIs). Although sometimes used interchangeably, it is important to understand that PIs and CIs do not convey the same information.

- *Confidence Interval*: Given the statistical parameter, such as the data population mean, the CI shows the likely range of values associated with this parameter [56, 137]. Therefore, the CI for the mean of the target probability distribution conveys no information regarding the dispersion of values around the mean.
- *Prediction Interval*: The PI provides a possible range of values for a future individual observation to be made. Unlike a CI, the PI is not predicting the mean, but instead the individual value of the target sample [137]. Therefore, a much greater uncertainty is associated with a PI than a CI, which is why a PI always delivers a wider range of values than a CI.

Based on the definition, it should be evident that the use of a CI allows for the uncertainty of a parameter to be quantified, given a set of point predictions. For example, if the MAPE (or NRMSE) of a set of predictions is obtained, then the CI delivers a range of MAPE values that can be expected. In other words, should a MAPE metric be provided to decision makers, instead of providing a single MAPE value, it is possible to deliver a MAPE value, together with a degree of confidence.

For this research, uncertainty is quantified in the form of CIs with the use of the Bootstrapping technique. Bootstrapping is a non-parametric (assumes no distribution) method, which allows for the estimation of the sampling distribution of almost any statistic [138]. The process of estimating the probability distribution of a random variable $R(B, X)$ starts with a new data set, randomly sampled with replacement from the original parent distribution $\boldsymbol{X}$ of n observations. This new data set sample is referred to as the bootstrap sample $\boldsymbol{B} = (b_1, b_2, ..., b_n)$. This process of acquiring bootstrap samples $\boldsymbol{B}$ is repeated m times, with m as a large number (1000 to 10 000). With m bootstrap samples obtained, the mean for each bootstrap sample is obtained, $(\bar{\boldsymbol{B}}_{\boldsymbol{1}}...\bar{\boldsymbol{B}}_{\boldsymbol{m}})$, referred to as the bootstrap estimates. With a total of m bootstrap estimates the values are sorted in ascending order and a probability distribution is obtained, from which the Bootstrap CI is defined. This Bootstrap CI ultimately defines by how much the mean varies across the samples.

For this research, m is set to 10 000 with the MAPE as statistic, so that 10 000 bootstrap estimates of the MAPE are obtained. With the bootstrap estimates sorted in ascending

order, the 95 % CIs are obtained as all of the values between the first 2.5 % and last 2.5 % values. This entire process is visually illustrated with Figure 4.1. The bootstrap CI algorithm implementation is demonstrated with Algorithm 1.

For the reasons mentioned above in Section 4.2.1, the MAPE serves as final metric of evaluation. With the bootstrap estimates obtained together with the lower and upper bounds of the CI, the probability of all possible values of the estimated MAPE is defined. The bigger the difference between the lower and upper bound values, the wider the overall dispersion of values, which is indicative of a higher uncertainty.

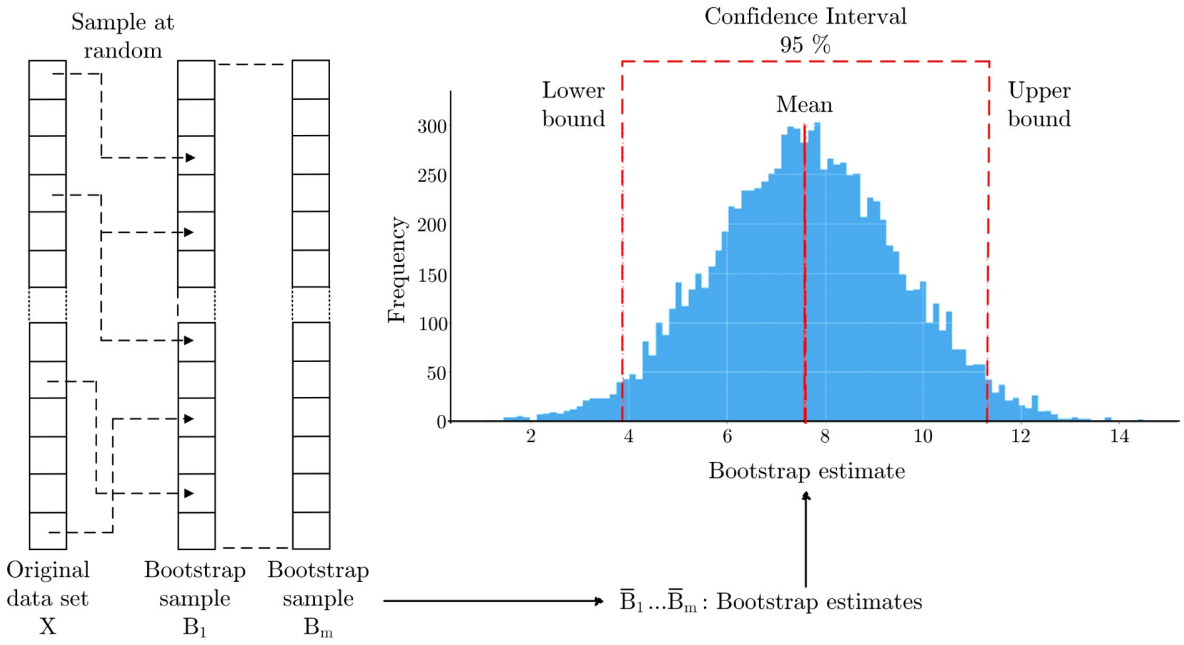


Figure 4.1: Example of the bootstrap process to obtain a 95% confidence interval.

Algorithm 1 Determine 95 % confidence intervals with Bootstrap analysis.

Let $\hat{y}$ be vector of predicted power values
Let y be vector of measured power output values
bootstrap_ estimates ← initialise empty list
a ← 95 % confidence interval
n ← number of y samples
m ← 10 000
y_diff ← $\hat{y}$ - y
for m **do**
    bootstrap_ sample ← sample n data points at random from y_diff
    bootstrap_ estimate ← calculate $MAPE$ from *bootstrap_ sample*
    save *bootstrap_ estimate* to *bootstrap_ estimates* list
end for
sort ascending *bootstrap_ estimates*
determine *lower bound* and *upper bound* values with a as criteria
`Return` *bootstrap_ estimates*, *lower bound*, *upper bound*

Given this information, an example application towards decision making would be where high impact decisions are made considering a 95 % CI (wider range of MAPE values), whereas low impact decisions might only require a 50 % CI (narrower range of values).

4.3 Macro-level PV-system forecasts

4.3.1 Overview

With the objective of delivering conventional macro-level forecasts, using state-of-the-art models, the process of training these models and the results obtained are presented in this section. Importantly, the obtained macro-level forecasting accuracies serve as the baseline for this research, to which the proposed methodology of aggregated inverter-level forecasts is compared. For convenience, the model development framework is once again illustrated here with Figure 4.2

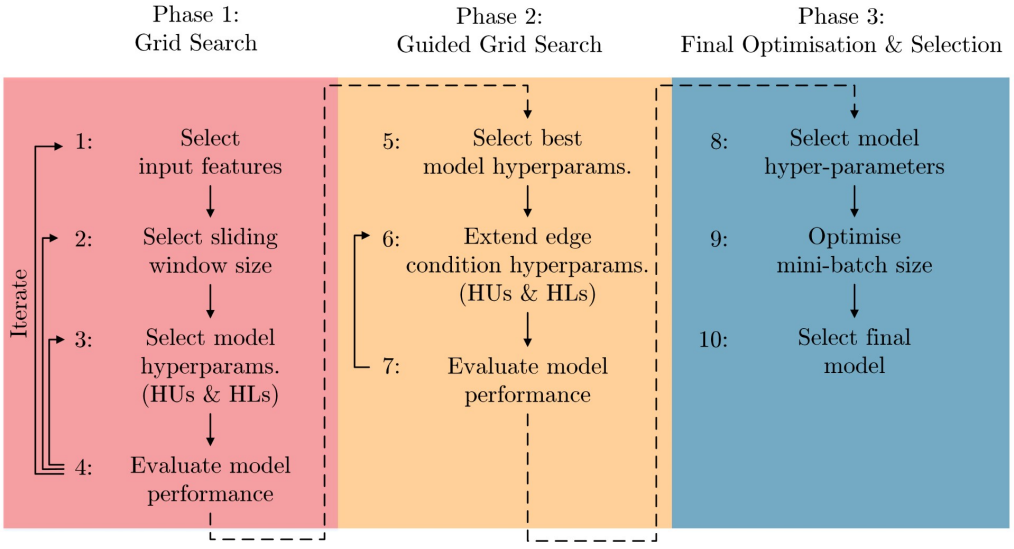


Figure 4.2: Macro-level forecast model development framework.

4.3.2 Phase-1 development

Guided by the framework presented in Figure 4.2, three phases of model development are completed. With Phase-1, various input features, hyperparameters and sliding-window sizes are explored for each model (FFNN, LSTM-RNN, GRU-RNN). The initial solution space for every DL model is empirically determined from an extensive search process, which consists of a diverse set of sampled combinations of HUs, HLs and MB sizes. From this empirical analysis a widespread starting solution space is formally defined, as presented in Table 4.1.

Important to mention from Table 4.1 is the selection of hyperparameter sizes to the power of two (x2). This is due to the improved runtime seen with models trained on Graphics Processing Units (GPUs) [108]. During model development, the HL width (i.e. the number of HUs) is also kept constant so that each HL has the same number of selected HUs applied. Constant HL width is selected for model development, instead of also iterating over different numbers of HUs in each HL, since the latter approach would drastically increase the required number of iterations. Instead, by keeping HL width constant, the large initial grid-search solution space, combined with various sliding-window sizes and input feature combinations provides more than enough opportunity for models to be adequately optimised.

Table 4.1: Phase-1 initial grid-search solution space defined for each model.

Item	Variables
Input features historic	P, GHI, T, WS, WD, RH, AP
Input features future	month, β, Φ_S
Sliding window sizes [h]	1, 2, 3, 6, 24, HISIMI + xh
MB sizes	32, 64
FFNN	
HLs	2, 3
HUs per HL	32, 64, 128, 256, 512, 1024
LSTM-RNN	
HLs	2, 3
HUs per HL	16, 32, 64, 128, 256, 512
GRU-RNN	
HLs	2, 3
HUs per HL	16, 32, 64, 128, 256, 512

Before the analysis continues, important to note is that the Phase-1 results in Tables 4.2 - 4.4, represent the validation-set results. Also, although the historic input features are used in various combinations, the month and sun position angles (Φ_S, β) associated with the forecast horison are provided as input features for every iteration. Reason being that these three features provide information regarding time and seasonal variations, which has proven to be crucial for good model performance.

Regarding the HISIMI + xh sliding window iterations, the selection of x is made according to the sliding window that performed best, given the input feature combinations. For example, the macro-level FFNN Model A results in Table 4.2, with P, GHI and T as input features, delivered the most accurate forecasts with a sliding window of 1 h. Thus, for the HISIMI + xh sliding window iteration, x h is selected as 1 h.

With the solution space defined, the input features are progressively explored, which results in seven unique combinations of historic input features applied to each DL model. This is demonstrated with the Model A to Model G iterations in Tables 4.2 - 4.4. Model training for all three DL models starts with P, GHI and T as principal input features. The reason being the strong relationship of these variables with power output, as evident from published literature (Tables 1.3 - 1.4), the correlation analysis (Appendix A.5.3) and experimentation.

Interesting from the results of all three Tables 4.2 - 4.4, is that the addition of WS, shown with Model B, fairly maintained or improved model accuracy compared to the Model A iterations. However, the addition of WD as input, reduced overall model accuracy for all three macro-level models, as shown with the Model C results. This reduction in model accuracy can be attributed to added complexity and the lower correlation of WD with power output. Importantly, these results do not contradict the observations in Section 1.3 regarding the influence of wind on inverter power output and overall PV system dynamics. These results merely indicate model preference for input features, with preference being a trade-off between both model accuracy and model complexity.

With the results obtained for each of the Model A iterations, the P, GHI and T as historic input features have proven to deliver better forecasting accuracy than the Model B and C iterations, for all three DL models. Thus, the next feature combinations applied are the additions of RH and AP. The accuracies obtained for these feature combinations are presented with the Model D, Model E and Model F results. Evident from Tables 4.2 - 4.4 is

that the combination of P, GHI, T and AP (Model E) as input features delivers good results for all three macro-level models. However, similar to WD, the addition of RH yields no increase in accuracy for any of the macro-level models, as evident from the Model D results.

Since the addition of AP and WS delivered reasonable accuracy, the final set of feature combinations consist of P, GHI, T, AP and WS as input of the Model G iterations. These models show similar performance accuracies for the FFNN and LSTM-RNN models, across all sliding window iterations as compared to the individual use of AP or WS together with the P, GHI and T features. However, the GRU-RNN models did not perform well with the addition of AP and WS. These results support the remarks made in Section 3.4, where it was mentioned that no generic formula exists regarding the optimal selection of input features.

Table 4.2: Average validation set NRMSE results obtained for FFNN macro-level model forecasts as evaluated for specific input features and all weather conditions.

	NRMSE [%]						
sliding window	**Model A**	**Model B**	**Model C**	**Model D**	**Model E**	**Model F**	**Model G**
1 h	**8,24**	8.26	8.43	8.35	8.25	8.37	8.28
2 h	8.25	8.28	8.45	8.46	8.36	8.40	8.29
3 h	8.25	8.31	8.55	8.37	8.25	8.45	8.34
6 h	8.26	8.37	8.66	8.51	8.26	8.50	8.43
24 h	8.45	8.44	8.93	8.60	9.43	9.99	8.51
HISIMI + x h	8.31	8.32	8.49	8.39	8.32	8.43	8.33
Input features	P,GHI,T	P,GHI,T, WS	P,GHI,T, WS,WD	P,GHI,T, RH	P,GHI,T, AP	P,GHI,T, AP,RH	P,GHI,T, AP,WS
Future input features	month of year, solar angles Φ_S, β						

Table 4.3: Average validation set NRMSE results obtained for LSTM-RNN macro-level model forecasts as evaluated for specific input features and all weather conditions.

	NRMSE [%]						
sliding window	**Model A**	**Model B**	**Model C**	**Model D**	**Model E**	**Model F**	**Model G**
1 h	8.41	8.41	8.56	8.57	8.39	8.54	8.40
2 h	8.46	8.45	8.65	8.61	8.42	8.54	8.46
3 h	**8.36**	8.41	8.67	8.63	8.37	8.66	8.44
6 h	8.39	8.45	8.67	8.49	8.41	8.59	8.41
24 h	8.49	8.42	8.60	8.46	8.39	8.46	8.46
HISIMI + x h	8.55	8.57	8.74	8.65	8.47	8.74	8.53
Input features	P,GHI,T	P,GHI,T, WS	P,GHI,T, WS,WD	P,GHI,T, RH	P,GHI,T, AP	P,GHI,T, AP,RH	P,GHI,T, AP,WS
Future input features	month of year, solar angles Φ_S, β						

Table 4.4: Average validation set NRMSE results obtained for GRU-RNN macro-level model forecasts as evaluated for specific input features and all weather conditions.

	NRMSE [%]						
sliding window	**Model A**	**Model B**	**Model C**	**Model D**	**Model E**	**Model F**	**Model G**
1 h	8.27	8.25	8.38	8.37	8.28	8.25	8.31
2 h	8.27	8.26	8.34	8.34	8.24	8.24	8.37
3 h	8.24	8.24	8.38	8.35	8.23	8.20	8.34
6 h	8.18	8.20	8.29	8.37	**8.16**	8.20	8.24
24 h	8.25	8.25	8.38	8.28	8.24	8.28	8.31
HISIMI + x h	8.27	8.22	8.37	8.30	8.26	8.25	8.31
Input features	P,GHI,T	P,GHI,T, WS	P,GHI,T, WS,WD	P,GHI,T, RH	P,GHI,T, AP	P,GHI,T, AP,RH	P,GHI,T, AP,WS
Future input features	month of year, solar angles Φ_S, β						

4.3.3 Phase-2 and Phase-3 development

With this section the execution of Phase-2 and Phase-3 is presented for the FFNN, LSTM-RNN and GRU-RNN macro-level models. Firstly, Phase-2 of model development consists of identifying the most probable model solutions (identified from Phase-1 results) and further expanding on the hyperparameter grid-search domain. Specifically, the aim is to identify and expand on the best model solutions, for which the HU or HL hyperparameter selections are on the edge of the explored grid-search domain. This coordinated expansion of the grid-search solution space is executed as defined in Section 3.6.2.

Once Phase-2 is completed and the input feature combinations, sliding-window size and the best model architecture is established, Phase-3 serves as the final step towards model optimisation. As motivated in Section 3.6.2, the MB sizes applied are limited to only 32 and 64 for all Phase-1 and Phase-2 iterations. From an empirical analysis of initial model exploration, these MB sizes were identified as a good selection. With the use of the Adam optimiser and the LR also kept constant at $1e^{-4}$, these smaller MB sizes together with the small LR, further allow for good regularising effects so that solutions generalise well. Also, apart from the various iterations of input feature selection, it is observed that the largest performance gains are typically achieved with changes in architecture (HU, HL). However, to ensure that the full spectrum of solutions are considered, Phase-3 serves as the final optimisation step, with additional MB sizes explored.

4.3.3.1 FFNN

Based on the preliminary FFNN results in Table 4.2, Model iterations A, B, E and G are identified as the best models to proceed with for Phase-2 optimisation. In Table 4.5 a summary regarding the hyperparameter values is provided, as identified for the best performing models of Phase-1 for the FFNN model development. Given the initial FFNN solution space of Table 4.1, only FFNN Model A delivered an edge case solution, with HL=3. The most accurate model hyperparameter selections of Models B, E and G delivered no edge case solutions, showing no need for further hyperparameter optimisation. Therefore, the hyperparameter solution space of FFNN Model A is expanded to HL=4 and combinations of 32, 64 and 128 HUs for each HL. With the MB sizes maintained at 32 and 64, two iterations are effectively executed for each of these HU and HL combinations.

Table 4.5: Averaged NRMSE results of best FFNN macro-level models for Phase-1 and Phase-2 model development.

		Best model solutions			
		Model A	**Model B**	**Model E**	**Model G**
Development Phase	Historic Input features	[P,GHI,T]	[P,GHI,T,WS]	[P,GHI,T,AP]	[P,GHI,T,AP,WS]
	Future input features	month of year, solar angles Φ_S, β			
Phase - 1	sliding window	1 h	1 h	1 h	1 h
	HLs	3	2	2	2
	HNs	[64,64,64]	[128,128]	[128,128]	[128,128]
	NRMSE [%]	**8.24**	8.26	8.25	8.28
Phase - 2	sliding window	1 h			
	HLs	4			
	HNs	[64,64,64,64]			
	NRMSE [%]	8.26			

With the solution space expanded for the Phase-2 results, model accuracy did not improve. Thus, the most accurate overall FFNN model is identified for HL=3 and HU=64. Model development is finalised with the execution of Phase-3, where additional MB sizes of 4, 8, 16, 128 and 256 are explored. From these results, presented in Table C.1 in Appendix C.1, a MB size of 64 is proven as the optimal selection.

4.3.3.2 LSTM-RNN

With forecasting accuracies obtained for the initial LSTM-RNN Phase-1 solutions space, the iterations of Models A, B, E and G in Table 4.3 are also identified as probable solutions to further expand on. The hyperparameters of the best models identified for Phase-1 are presented in Table 4.6.

From Table 4.6, the hyperparameter selections of all four LSTM-RNN models are edge conditions of the initial grid-search domain, with HL=3 and HU=16. Therefore, with Phase-2 of model development executed, the HUs are firstly reduced to HU=8, with the HL maintained at HL=3. This is followed by the expansion of the solution space with HL=4 and the HUs explored within a range of 8, 16 and 32. Once again, all model iterations are performed for the MB sizes of 32 and 64. With these new model architectures applied, the best Phase-2 results obtained are presented in Table 4.6.

From these results, the Phase-2 iterations do not improve on the results of the original Phase-1 solution space. Therefore, LSTM-RNN Model A is selected as the superior solution, with HL=3 and HU=16, with a sliding-window of 3 h.

With Phase-3 as the final step for model development, additional MB sizes of 4, 8, 16, 128 and 256 are explored. The results obtained concerning all of the MB sizes are presented in Table C.2 in Appendix C.2. From these results, a MB size of 32 is proven as the optimal selection.

4.3.3.3 GRU-RNN

The third and final macro-level forecasting model applied is the GRU-RNN model. With Phase-1 of model development completed the input feature combinations, which delivered the best results are displayed in Table 4.7.

Table 4.6: Averaged NRMSE results of best LSTM-RNN macro-level models for Phase-1 and Phase-2 model development.

		Best model solutions			
		Model A	**Model B**	**Model E**	**Model G**
Development Phase	Historic input features	[P,GHI,T]	[P,GHI,T,WS]	[P,GHI,T,AP]	[P,GHI,T,AP,WS]
	Future input features	month of year, solar angles Φ_S, β			
Phase - 1	sliding window	3 h	1 h	3 h	1 h
	HLs	3	3	3	3
	HUs	[16,16,16]	[16,16,16]	[16,16,16]	[16,16,16]
	NRMSE [%]	**8.36**	8.41	8.37	8.4
Phase - 2	sliding window	3 h	1 h	3 h	1 h
	HLs	4	4	4	4
	HUs	[8,8,8,8]	[8,8,8,8]	[32,32,32,32]	[8,8,8,8]
	NRMSE [%]	8.37	8.43	8.54	8.41

Identical to the performances of the FFNN and LSTM-RNN models, the GRU-RNN combinations of Models A, B, E and G all delivered the highest accuracies. With edge case conditions identified with HL=3, the Phase-2 solution space is firstly expanded with HL=4 and the number of HUs per layer set to 8, 16 and 32 for the Model A and B iterations. The number of HUs for the expansion of Model G is set to 16, 32 and 64 per layer. With the MB sizes of 32 and 64 once again maintained, the results delivered from the Phase-2 hyperparameter iterations are presented in Table 4.7.

Important to note is that GRU-RNN Model A delivered a slight increase in forecasting accuracy for the expansion of HL=4. Therefore, in accordance with the Phase-2 guided grid-search strategy, another expansion of the hyperparameter solution space is executed, with HL=5 and HUs set to 8, 16 and 32 for each layer once again. Since, this expansion of the hyperparameter solution space does not deliver higher accuracies, GRU-RNN Model E evidently delivers the highest forecasting accuracy, with HL=2 and HU=64.

Table 4.7: Averaged NRMSE results of best GRU-RNN macro-level models for Phase-1 and Phase-2 model development.

		Best model solutions			
		Model A	**Model B**	**Model E**	**Model G**
Development Phase	Historic Input features	[P,GHI,T]	[P,GHI,T,WS]	[P,GHI,T,AP]	[P,GHI,T,AP,WS]
	Future input features	month of year, solar angles Φ_S, β			
Phase - 1	sliding window	6 h	6 h	6 h	6 h
	HLs	3	3	2	3
	HNs	[16,16,16]	[16,16,16]	[64,64]	[32,32,32]
	NRMSE [%]	8.18	8.20	**8.16**	8.20
Phase - 2 A	sliding window	6 h	6 h	–	6 h
	HLs	4	4	–	4
	HNs	[16,16,16,16]	[16,16,16,16]	–	[64,64,64,64]
	NRMSE [%]	8.17	8.24	–	8.20
Phase - 2 B	sliding window	6 h	–	–	–
	HLs	5	–	–	–
	HNs	[8,8,8,8,8]	–	–	–
	NRMSE [%]	8.20	–	–	–

With Phase-3 as final optimisation step, the MB size is evaluated for the additional values of 4, 8, 16, 128 and 256. From these final results obtained, as provided in Table C.3 in Appendix C.3, a MB size of 64 is proven as the optimal selection.

4.3.4 Macro-level model development summary

With the development of the FFNN, LSTM-RNN and GRU-RNN macro-level models completed, historic P, GHI and T as input features were shown to have a high correlation with power output and delivered good results for all three models. It was further observed, that the addition of AP and WS as input features also proved to be valuable input features for all three macro-level models. However, the inclusion of RH and WD was found to reduce model accuracies overall, which was attributed to a low correlation with power output data and an increased model complexity due to the addition of these features.

With all three phases of model development completed, as guided by the model development framework of Section 3.6.2, it can be stated with confidence that the extensive training of all three macro-level models has been executed with an unbiased amount of effort. The final hyperparameter selections and input data preferences, which delivered the highest forecasting accuracies for each model are summarised in Table 4.8.

Table 4.8: Best macro-level PV-system forecasting model details.

	Model		
	FFNN	**LSTM-RNN**	**GRU-RNN**
HLs	3	3	2
HUs	[64, 64, 64]	[16, 16, 16]	[64, 64]
MB size	64	32	64
Sliding window	1 h	3 h	6 h
Training time [min]	5.06	25.72	11.48
Historic input features	[P, GHI, T]	[P, GHI, T]	[P, GHI, T, AP]

From Table 4.8, the longer 3 h and 6 h sliding window preferences of the LSTM-RNN and GRU-RNN models are in contrast to the shorter 1 h historic sliding widow preference of the FFNN model. This supports the model theory presented in Section 3.3, which discussed the ability of the RNN-based models to prioritise input data, without being overwhelmed like by the number of time-steps fed into the model as input. Also interesting to note is that only the GRU-RNN model performs better with AP as input feature, in addition to P, GHI and T as input data.

4.3.5 Results

With the extensive model development processes completed, the forecasting ability of each multi-step macro-level model is demonstrated with Figure 4.3 for the 1, 3 and 6 h forecast intervals. As mentioned, with forecasts delivered at 15 min resolution for the forecast horison of 1 - 6 h ahead, each model effectively predicts 21 time-steps ahead. However, only the 1, 3 and 6 h ahead time-series graphs are presented here, since these time-steps deliver a good overall demonstration of the progressive change in the multi-step forecasting accuracies.

Also, with a test set length of 1 year, the data segment presented with Figure 4.3 is specifically shown since it demonstrates forecasting scenarios, which are all representative of various (and difficult) weather conditions. These weather conditions (day types) are defined in the Figure 4.3 caption. From these graphs the difficulty level of delivering accurate forecasts

evidently increases, as the forecast horison and variability of the environment increases. For a more complete perspective on the time-series prediction capability of these models, multiple forecasts for each day type are presented in Appendix C.4.

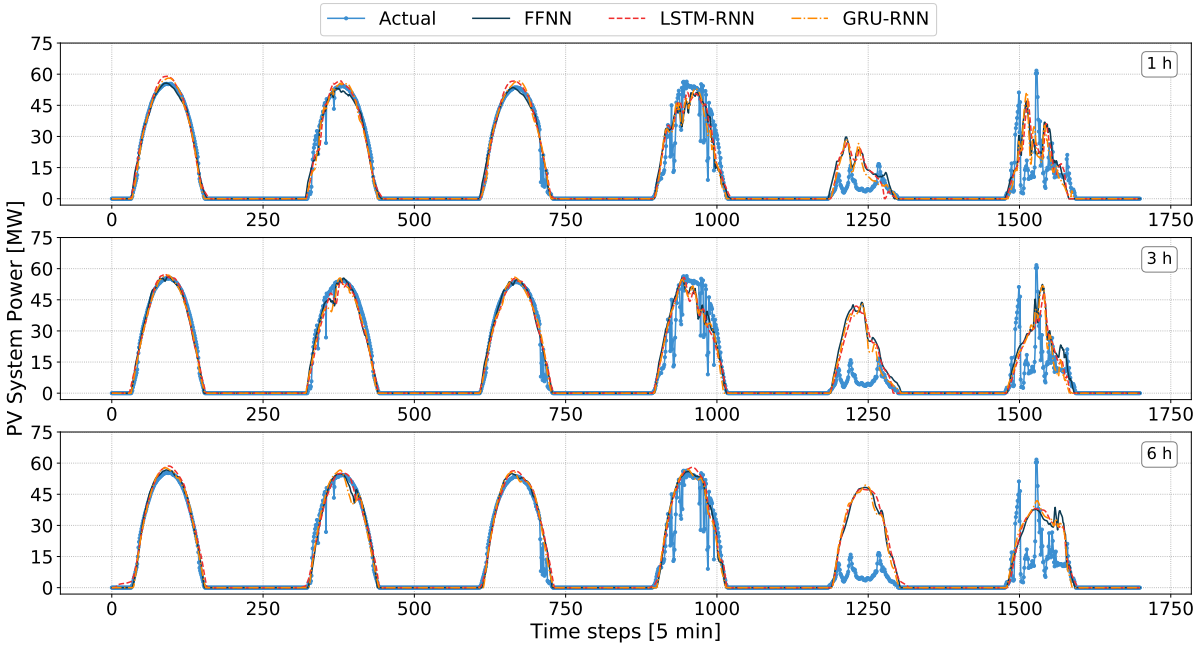


Figure 4.3: Time-series forecasts of macro-level models for 1 h, 3 h and 6 h ahead. Day-types from left to right: clear, clear-intermittent, clear-intermittent, intermittent, overcast and overcast.

The averaged NRMSE and MAPE results for the multi-step forecasts of 1 - 6 h ahead are summarised in Table 4.9. From these results the GRU-RNN model delivers the most accurate forecasts on average for all weather conditions. For clear day types, the FFNN is outperformed by the LSTM-RNN, but has similar NRMSE results for the overcast and intermittent day predictions. However, for the clear-intermittent days the FFNN presents a higher forecasting accuracy with NRMSE and MAPE values of 5.1 % and 2.24 %, as compared to the 5.38 % and 2.45 % of the LSTM-RNN. From these macro-level forecast results, the FFNN in general outperforms the LSTM-RNN.

Table 4.9: PV-system macro-level forecast results for all models and weather conditions as determined from the test set.

	Models			
	FFNN	**LSTM-RNN**	**GRU-RNN**	**Weather type**
NRMSE [%]	3.52	3.40	3.04	Clear
	5.10	5.38	5.01	Clear-Intermittent
	10.51	10.49	10.46	Intermittent
	14.98	14.98	14.34	Overcast
MAPE [%]	1.80	1.75	1.54	Clear
	2.24	2.45	2.17	Clear-Intermittent
	4.94	4.96	4.89	Intermittent
	8.08	7.97	7.72	Overcast
NRMSE [%]	8.19	8.23	**8.12**	All
MAPE [%]	3.53	3.61	**3.42**	All

Further context of the forecasting accuracies is presented with Figures 4.4 and 4.5, which respectively shows a heatmap of the NRMSE and MAPE values for each prediction point. From this complete demonstration of forecast accuracies, it is once again evident how overall forecasting accuracy declines with an increase in forecast horison and weather variability.

From the NRMSE values in Figure 4.4, the GRU-RNN model delivers the highest forecasting accuracies for the entire forecast horison, across all day types. This analysis further shows that, as the forecast horison increases beyond the 5 h mark for clear days, the FFNN outperforms the LSTM-RNN. These model dynamics are further accentuated in the MAPE heatmap results of Figure 4.5. From these MAPE heatmap results the improved performance of the FFNN model is further visible, as compared to the LSTM-RNN for clear-intermittent days. Finally, regarding the FFNN and LSTM-RNN macro-level model comparisons, the MAPE heatmap further shows how the FFNN delivers better accuracies for both the clear-intermittent and intermittent day types.

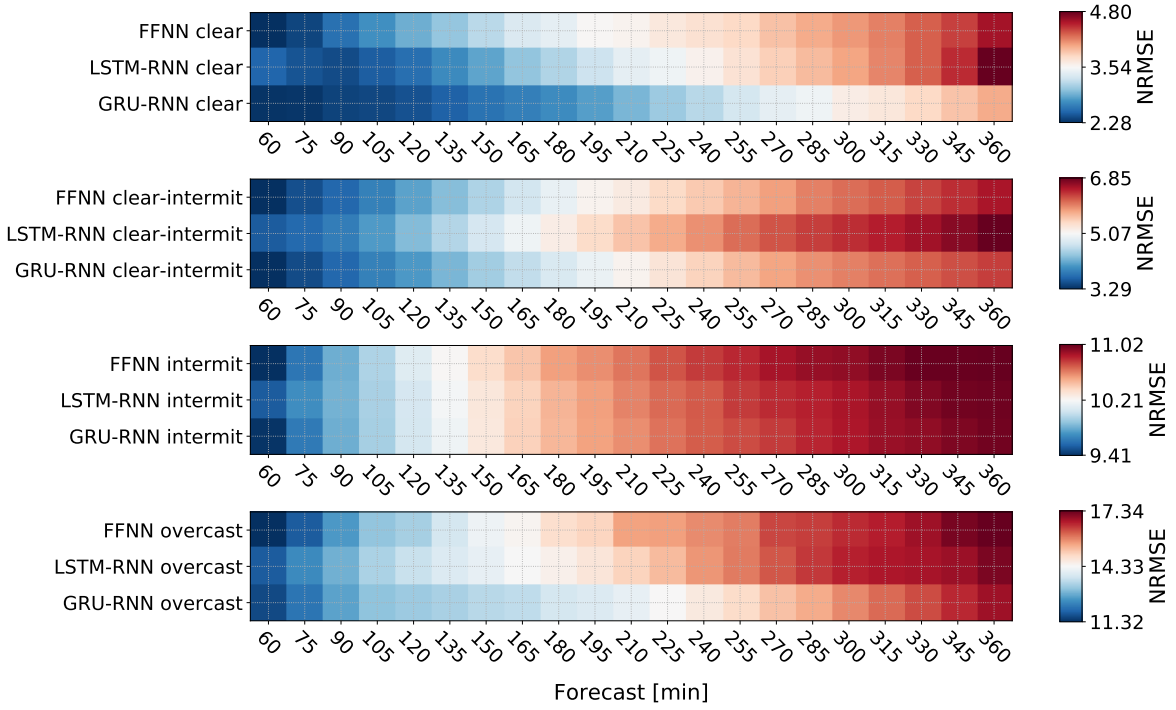


Figure 4.4: Test set NRMSE [%] results of macro-level forecast models as delivered for specific weather conditions.

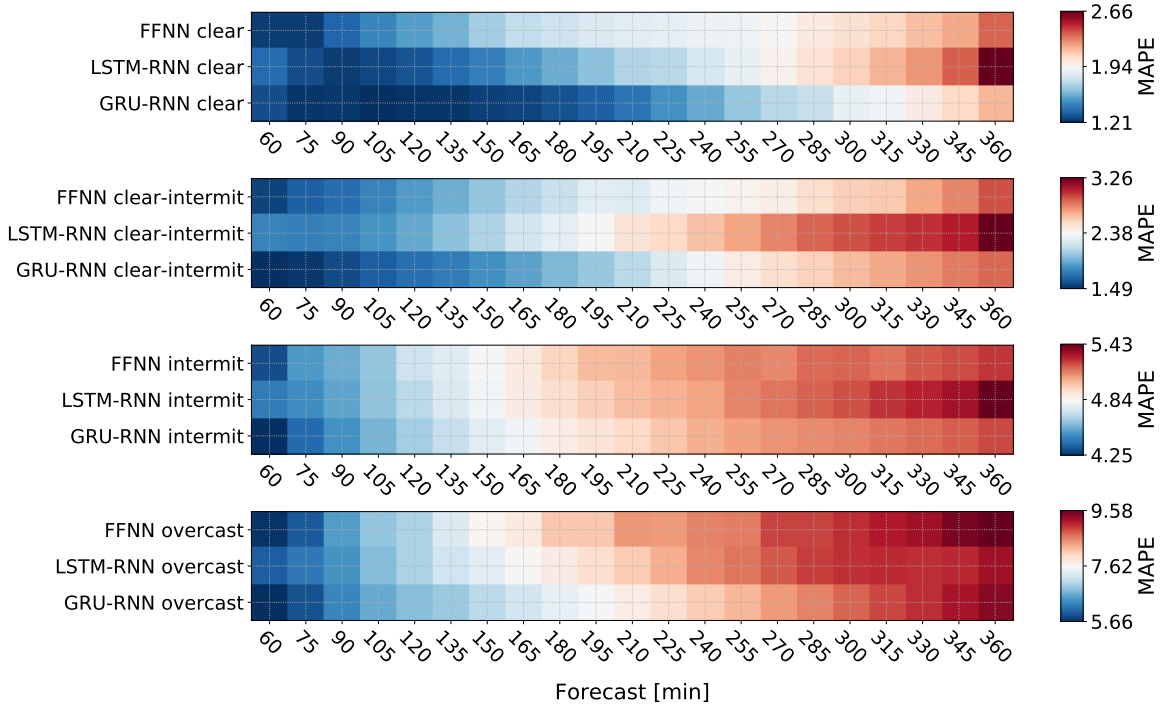


Figure 4.5: Test set MAPE [%] results of macro-level forecast models as delivered for specific weather conditions.

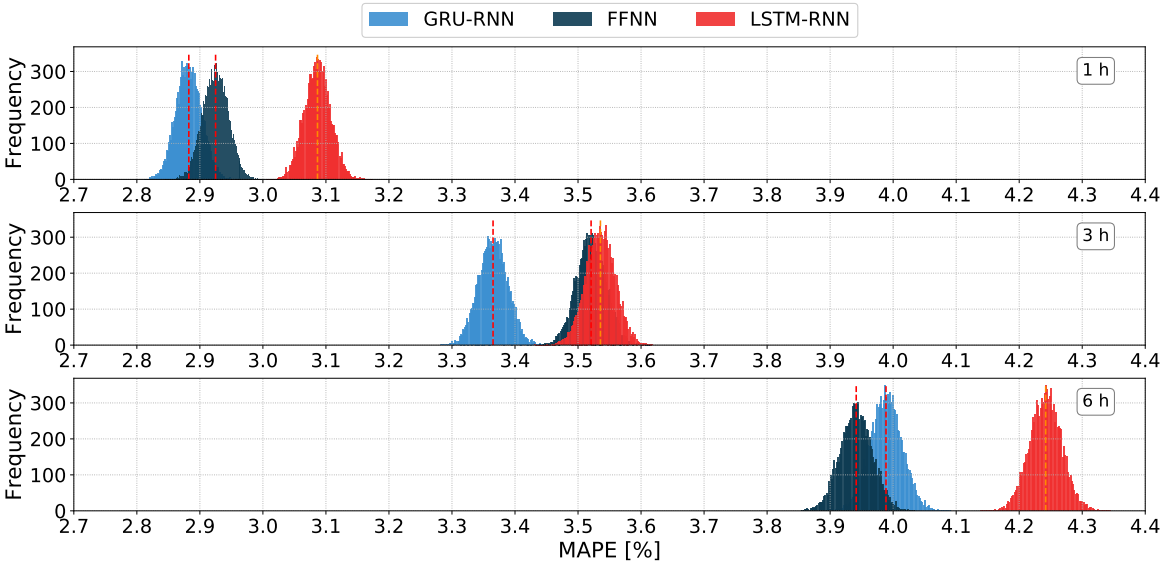


Figure 4.6: Test set MAPE [%] bootstrap analysis of the macro-level forecast models for all weather conditions.

For the reasons provided in Section 4.2, the MAPE serves as the decisive analysis metric regarding the forecasting accuracies. Given this context, the final evaluation of model performance is done with the Bootstrap analysis, based on the MAPE values. The results of this uncertainty analysis are presented in Figure 4.6, which demonstrates the average MAPE Bootstrap CI distributions for all weather conditions.

With the 1 h, 3 h and 6 h forecast intervals presented, the Bootstrap CI graphs clearly show the reduced forecasting accuracy and increase in uncertainty as the forecast horison increases. This is further evident from Table 4.10, which indicates how the CIs, represented by the base

of each distribution, also expand (less confidence) in width as the forecast horison increases. From Figure 4.6 it is further obvious that the GRU-RNN model outperforms the FFNN and LSTM-RNN models. The FFNN clearly also delivers a higher overall forecasting accuracy for the 1, 3 and 6 h ahead forecast horisons, as compared to the LSTM-RNN model.

Table 4.10: Test set 95 % Bootstrap CIs of the macro-level forecast models for all weather conditions.

All weather conditions				
Model	**Avg. MAPE [%]**	**Lower Bound [%]**	**Upper Bound [%]**	**CI width [kW]**
		1 h		
FFNN	2.92	2.89	2.97	59.74
LSTM-RNN	3.09	3.05	3.13	61.30
GRU-RNN	2.88	2.84	2.92	59.82
		3 h		
FFNN	3.52	3.47	3.57	68.45
LSTM-RNN	3.54	3.49	3.58	68.77
GRU-RNN	3.37	3.32	3.41	67.33
		6 h		
FFNN	3.94	3.89	3.99	74.48
LSTM-RNN	4.24	4.19	4.29	75.59
GRU-RNN	3.99	3.94	4.04	74.43

To summarise, with all of the macro-level results presented, it is concluded that the GRU-RNN model with a NRMSE of 8.12 % delivers the best overall accuracy, followed by the FFNN with a NRMSE of 8.19 %. These forecast accuracies are further supported by the deterministic heatmap results and Bootstrap CIs, which have delivered an average MAPE of 3.42 % and 3.53 % for the GRU-RNN and FFNN models, respectively.

4.3.6 Contextualising model accuracies

4.3.6.1 Baseline comparison

As a final measure to gauge model performance, the obtained macro-level forecasting results are compared to a Linear regression model, conventionally used in literature as a baseline for forecasting accuracy, as demonstrated in Figure 4.7.

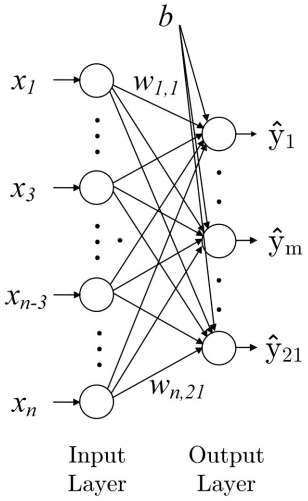


Figure 4.7: Demonstration of a basic fully connected Linear regression FFNN model.

To ensure that the best baseline model is obtained, an initial comparison is done with Lasso, Ridge and Linear regression models trained to deliver 1 - 6 h ahead forecasts. Lasso regression is simply the application of L1 regularisation applied with a normal Linear regression model, whereas Ridge regression refers to the use of L2-regularisation [108, pp.224-230]. The cost functions $J(\cdot)$ applied for the Linear, Ridge and Lasso regression models are presented with Eq. (4.3) - (4.5), respectively:

$$J(\boldsymbol{w}, \boldsymbol{b}) = \frac{1}{m} \sum_{i=1}^{m} \left(\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)} \right)^2 \tag{4.3}$$

$$J(\boldsymbol{w}, \boldsymbol{b}) = \frac{1}{m} \sum_{i=1}^{m} \left(\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)} \right)^2 + \lambda \boldsymbol{w}^2 \tag{4.4}$$

$$J(\boldsymbol{w}, \boldsymbol{b}) = \frac{1}{m} \sum_{i=1}^{m} \left(\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)} \right)^2 + |\lambda \boldsymbol{w}| \tag{4.5}$$

Where the $\boldsymbol{w}, \boldsymbol{b}$ and λ variables in Eq. (4.3) - (4.5) represent the neuron weights, bias terms and penalty term, respectively. Therefore, with Ridge regression defined by Eq. (4.4), the cost function is simply altered by multiplying the penalty with the square of the magnitude

of the neuron weights. Whereas, with Lasso regression (Eq. (4.5)), instead of taking the square of the weights (coefficients), magnitudes are taken into account. With λ serving as the penalty term, this is a hyperparameter to be manually selected during training.

These three models were trained with the exact model development framework used for the FFNN, LSTM-RNN and GRU-RNN models. However, the only difference being that instead of manually optimising for the number of HUs and HLs, the λ penalty term is optimised as hyperparameter. With all of the available input feature combinations and sliding-window sizes applied, the Linear regression model delivered the most accurate forecasts, compared to the Ridge and Lasso regression models.

With the best Linear regression model trained, the highest accuracy is achieved with a 24 h sliding window and P, GHI, T, AP and RH as input features. The 24 h sliding window is a clear indicator of the reduced representational power of the Linear regression model, since this model clearly relies strongly on a persistence approach. Also, the unique input feature combination once again further proves the suggestion that a high correlation analysis is not an indication of an ideal input feature combination set.

Regarding the forecasts, the time-series graph and test set MAPE results are demonstrated for the Linear regression (LinReg) model, in comparison to the obtained FFNN, LSTM-RNN and GRU-RNN macro-level models with Figures 4.8 and 4.9. With the MAPE bar graphs displayed for each forecast time step, the superior performances of the three DL-based models are evident. The overall LinReg model MAPE average recorded is 5.2 %, compared to the GRU-RNN average of 3.42 %.

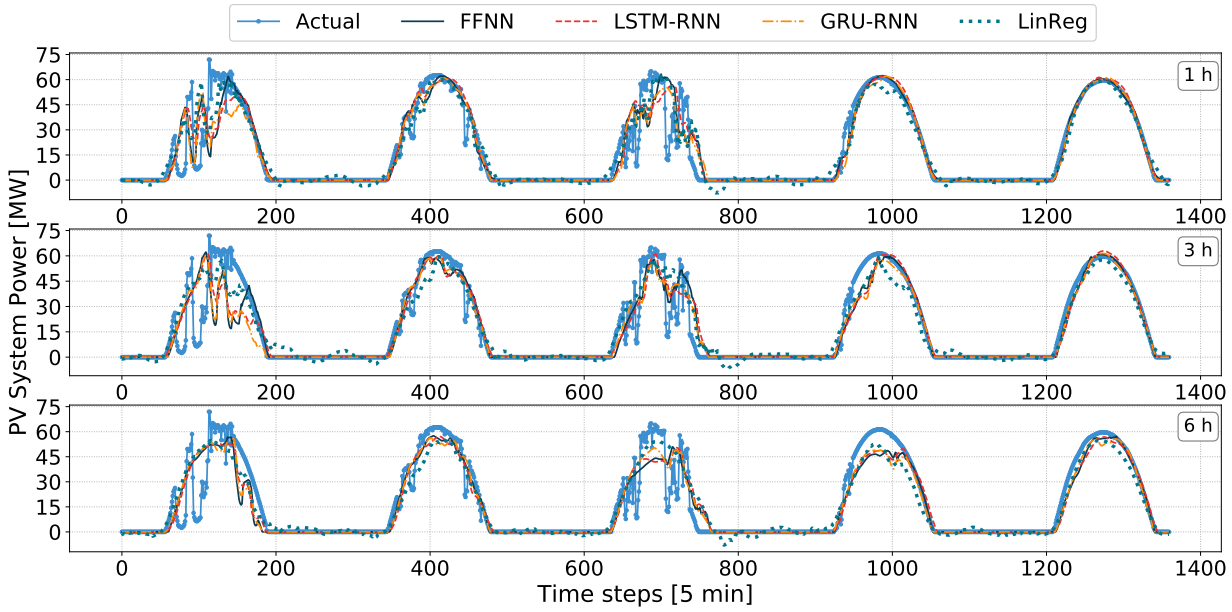


Figure 4.8: Time-series forecasts of the developed FFNN, LSTM-RNN and GRU-RNN models compared to a Linear regression baseline model.

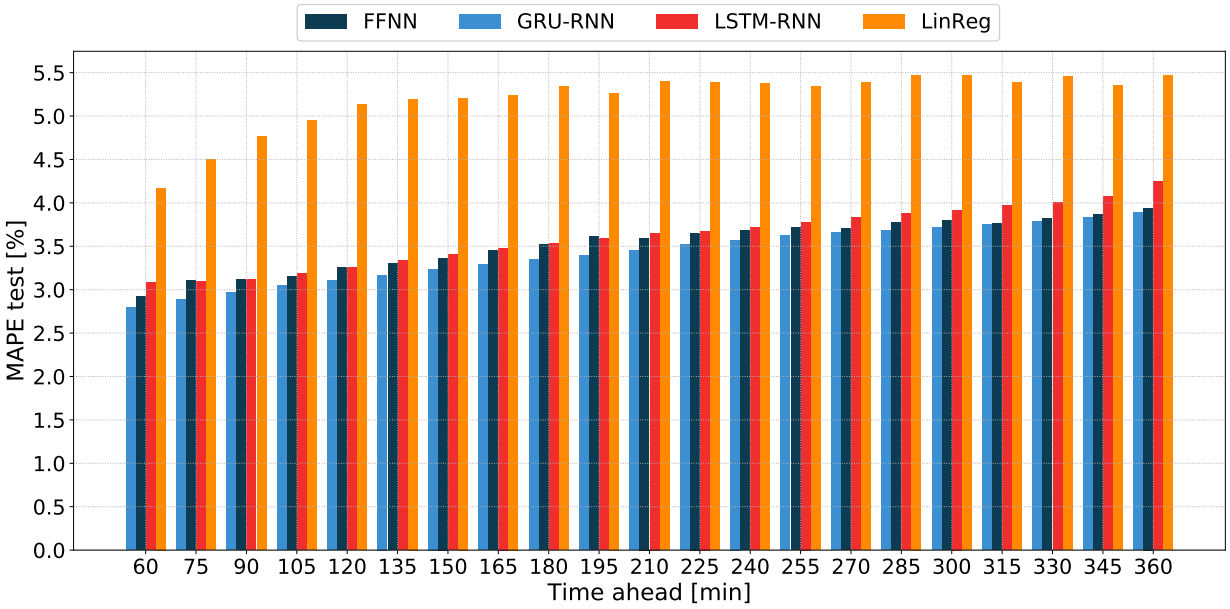


Figure 4.9: Averaged test set MAPE performance for all weather conditions as delivered for each time step by the FFNN, LSTM-RNN, GRU-RNN and Linear regression models.

4.3.6.2 Comparison of results to literature

As an indicator for more context regarding the accuracy and relevance of these model results, a comparison is made in Table 4.11 to the results of other published literature. The research studies compared are specifically selected, since these authors delivered results within the same research framework, for smaller PV systems. Thus, only historic on-site measured data was used to deliver intra-day forecasts, with the MAPE and NRMSE results normalised relative to PV system installed capacity. Finally, since the GRU-RNN model delivered the best macro-level forecasting results, it is selected for the comparison to literature results.

Table 4.11: Comparison of macro-level GRU-RNN model results to published literature.

Author	**Forecast**	**Accuracy**	
	Horizon	**Literature**	**GRU-RNN**
Y. Wu *et al.*		**NRMSE [%]**	**NRMSE [%]**
[9]		(Day-time only)	(Day-time only)
	1 h	System 1: 5.64	5.77
		System 2: 3.43	
		System 3: 6.57	
D. AlHakeem *et al.*		**MAPE [%]**	**MAPE [%]**
[56]	1 h	0.87 - 20.74	1.28 - 5.81
	3 h	2.06 - 36.64	1.16 - 7.69
	6 h	5.15 - 40.94	2.11 - 10.18
N. Sodsong *et al.*		**NRMSE [%]**	**NRMSE [%]**
[75]	1 h	9.64	7.08

Important to note is that this direct comparison, merely serves to give some context regarding a realistic expectation of the range of forecasting accuracies that can be expected, as has been obtained by published research. Any competitive claims of superior model performance

made would be very unscientific, since these results were generated for different PV systems, with different data sets.

4.4 Aggregated inverter-level forecasts

4.4.1 Model development framework

With the macro-level PV-system forecasts obtained for the FFNN-and RNN-based models, the next objective is to train and evaluate models for the aggregated inverter-level forecasts. Since the GRU-RNN model is proven as the most accurate macro-level RNN-based model, with the FFNN as the second most accurate model, the analysis is continued for these two (FFNN & GRU-RNN) models.

4.4.1.1 Input feature selection

With the purpose of delivering unbiased model comparisons, the inverter-level models are trained with the same model development framework applied for the macro-level models, as conveniently displayed again in Figure 4.10. However, with the ideal input features and sliding-window sizes obtained for the macro-level models, the question does present itself whether the computational effort of selecting a unique set of input features for the inverter-level models is necessary?

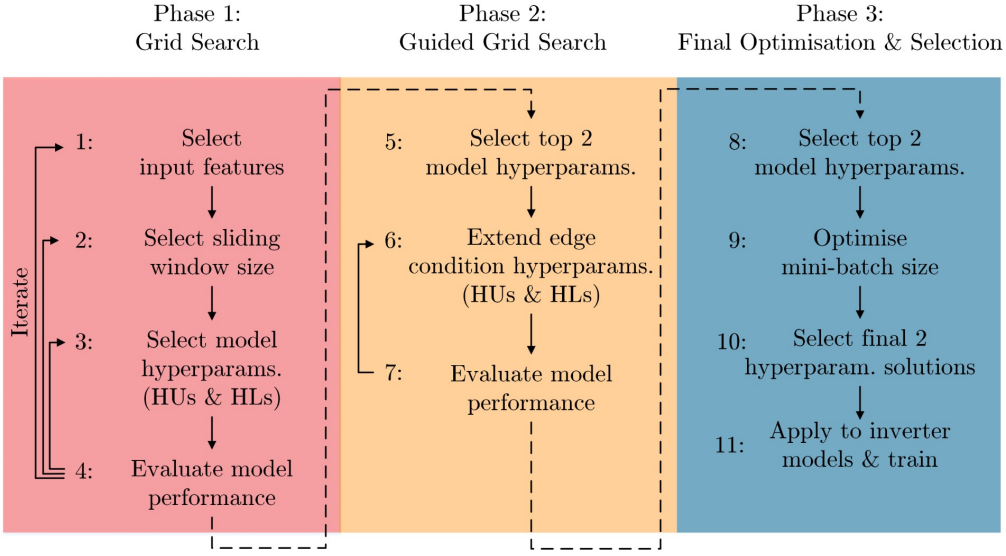


Figure 4.10: Forecast model development framework for aggregated inverter-level models.

The hypothesis that inspires this question, is that by training a DL model on macro-level power output data, essentially allows the trained model to holistically identify the input features, which have an overall correlation with the subsets of inverter PV modules. Thus, the holistic response of the overall PV-system, to a specific set of input features, should be representative of the same set of input features that are highly correlated with inverter-level behaviour.

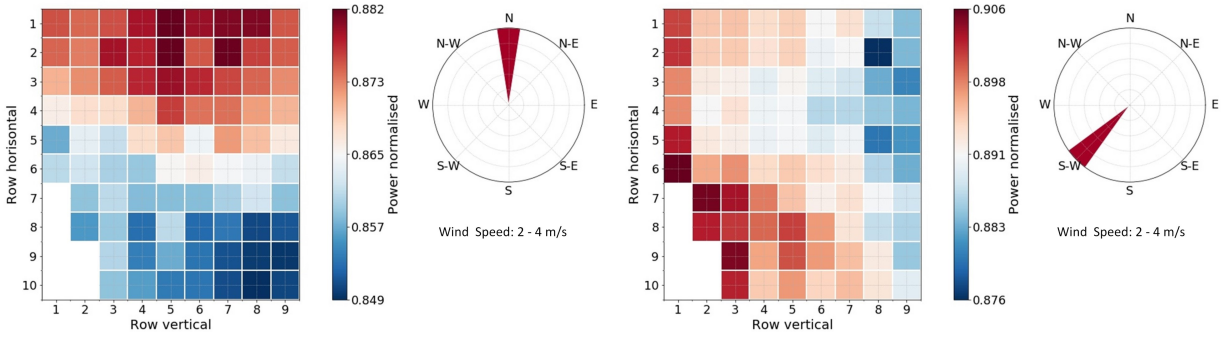
However, the counter argument to this hypothesis is that a subset of PV modules, smaller in size and at a specific location (with unique environment dynamics) cannot be assumed to have a similar preference for the same set of input features preferred by the macro-level models.

It is anticipated that there is some truth to both of these arguments regarding the selection of input features at inverter-level. This claim is based on the consideration that input features describing the atmosphere and environment can be categorised as:

- *High-level features*: Input features with a uniform distribution across the entire PV system, which are uninfluenced by PV-system layout and only start to differ across a large geographic expansion. For example, RH and AP are examples of variables, which generally have an equal level of exposure to the modules within large PV systems.
- *Medium-level features*: Input features, which in general should have a uniform distribution across the entire PV system, but are susceptible to a non-uniform distribution due to a dependence on other variables. Examples are GHI and T, which generally have a uniform distribution, but GHI is locally influenced by partial cloud cover and T is in turn influenced by both GHI and wind movement.
- *Low-level features*: These are input features, which can differ within a very small geographic spread, so that modules within large PV systems have a noticeable non-uniform exposure to these variables. Examples are WS and WD, which can significantly differ within the PV system.

This classification of input features can be used to the advantage of a reduction in computational expense concerning the inverter-level model development. For example, unlike wind movement, RH and AP remain unaffected by low-level influences and are regarded as uniform across the entire PV-system. With the macro-level models developed, RH was proven to reduce forecasting accuracy for all models, which automatically disqualifies the need to inspect this input feature at inverter level.

On the other hand, given the difference in the physical position of each inverter and the aerodynamic influence of the PV-system on wind movement, WS and WD as variables were proven to be very influential at inverter level. This influence on inverter-level power output dynamics is conveniently once again displayed with Figure 4.11.



(a) Power dynamics: North wind

(b) Power dynamics: South-West wind

Figure 4.11: Demonstration of the non-uniform effects of wind on the power output dynamics of the inverters.

4.4.1.2 Applied training methodology

Models are trained for each of the 84 inverters to accomplish the primary aim, which is the potential enhancement of PV forecasting accuracy using aggregated low-level forecasts. However, as previously mentioned, the execution of a full-stack grid-search approach to

obtain the optimal hyperparameter (HLs, HUs, MB size) selections would require a very high computational expense. In Section 3.6.3 it was suggested to rather cluster the inverters, with each inverter assigned to a group based on the similarity of the power output dynamics.

Guided by the results from the clustering analysis presented in Section 3.6.3, a total of 10 inverter clusters are assigned, as conveniently presented once again with Figure 4.12. For each of these clusters, a single inverter is selected as a representative. Thus, identical to the macro-level model development process, three phases of model development are executed for the training of each of these 10 representative inverters, optimising for hyperparameters and model parameters (step-1 in Figure 4.12).

Then, as demonstrated with step-2 in Figure 4.12, once the optimal hyperparameter selection is obtained for the representative inverters, these selections (HUs, HLs, MB size) are directly applied to each of the inverters in the corresponding cluster. For example, in Figure 4.12 the hyperparameter selection of representative inverter I67, is then applied directly to inverters I38, I40, I42, I44, I64, I65, I66, I68, I71, I72. Then, each of these inverter models are individually trained (parameter optimisation) to deliver the best forecasting solution, given the hyperparameters assigned.

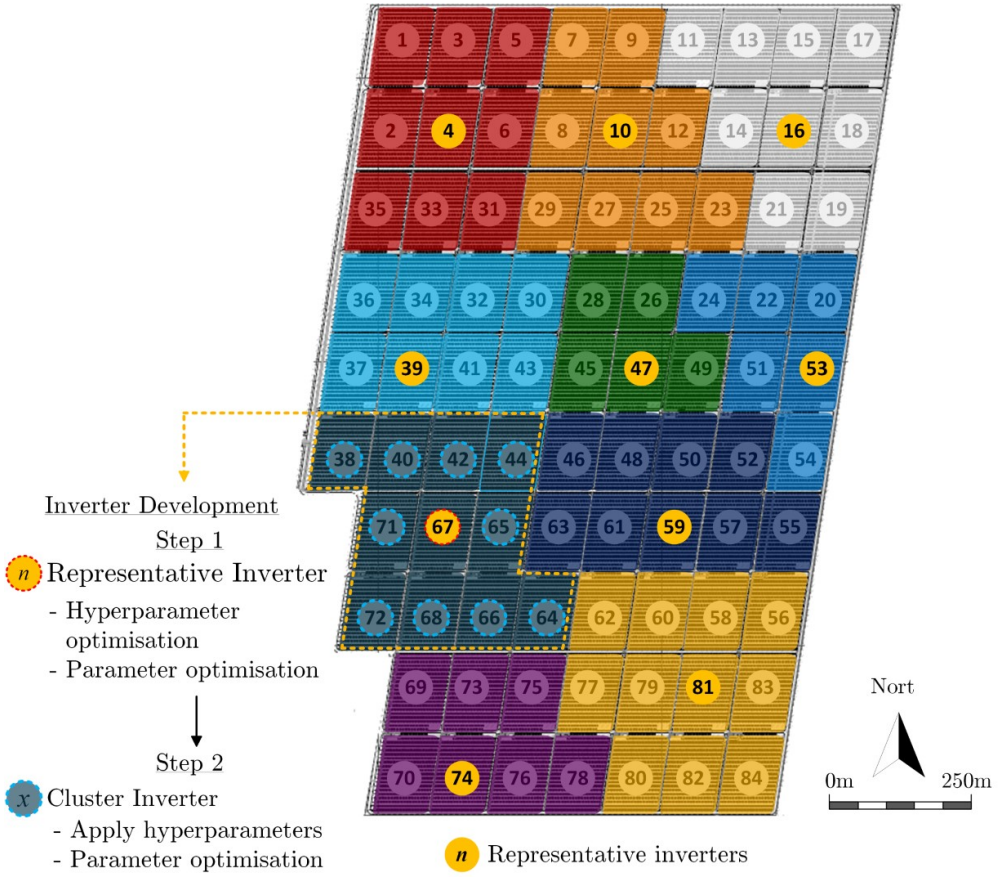


Figure 4.12: Inverter clusters with hyperparameter assignment visualised.

4.4.2 FFNN model development

For the development of the FFNN macro-level model, the P, GHI and T as historic input features, delivered the highest forecasting accuracy. To investigate whether these input features are also optimal for the inverter-level models, WS and WD are investigated as possible features that could further improve forecasting accuracy. The use of RH and AP will not be considered. The reason being that these high-level input features did not improve

the macro-level FFNN forecasting ability and are, therefore, also not expected to deliver any added value for inverter-level predictions as was explained in Section 4.4.1.1.

To ensure the optimal set of input features are used, a direct comparison is made regarding the aggregated inverter-level forecasts. The models compared are labelled as follows:

- *A*: Inverter-level FFNN model with input features: P, GHI, T
- *B*: Inverter-level FFNN model with input features: P, GHI, T, WS, WD

Phase-1 development

For Phase-1 of FFNN inverter-level model development, the same initial solution space is defined for the hyperparameters as used for the FFNN macro-level models. This Phase-1 grid-search solution space is defined in Table 4.12, for each FFNN inverter model.

Table 4.12: FFNN aggregated inverter-level model development Phase-1 solution space.

Model	HLs	HUs per HL	MB sizes	Sliding window	Input features
FFNN *A*	2, 3	32, 64, 128, 256, 512, 1024	32, 64	1 h	P, GHI, T
FFNN *B*					P, GHI, T, WS, WD

Important to note from Table 4.12 is the sliding-window size of 1 h. This might appear contrary to the development framework of Figure 4.10, where step-2 of Phase-1 requires an iteration of various sliding-window sizes. However, an interesting observation made from the training procedure of the macro-level FFNN models (Table 4.2), was that a sliding window of 1 h always delivered the most accurate (NRMSE values) forecasts, irrespective of the input feature combinations. To potentially exploit this observation, a preliminary investigation is done at inverter-level regarding the sliding-window size in an attempt to identify a universal sliding-window size. Should a specific sliding-window size prove to be the preference for different input feature combinations and inverters, this will help to avoid any unnecessary computational expense (crucial for real-world model development & deployment).

With a sample-set of representative inverters I10, I53 and I67, a full range of sliding-windows (1, 2, 3, 6, 24 h) are applied for the given sets of input features of the FFNN inverter-level models *A* and *B*. These three inverters (I10, I53, I67) are selected as the sample-set, since this represents sections within the PV system that greatly differ from one another regarding position and exposure to the immediate environment. From this sample-set of results, all three inverters have proven to be in agreement of the same sliding-window size of 1 h, for both sets of input features as shown in Table D.1 and Table D.2 in Appendix D.1.

Thus, Phase-1 is executed for models *A* and *B* (applied to all 10 representative inverters) with all of the hyperparameter combinations, for a fixed 1 h sliding-window size.

Phase-2 development

With Phase-1 completed for the 10 representative inverters, the architectures (HUs, HLs) that deliver the highest accuracies are presented in Table 4.13 and Table 4.14. Important to note is that the top two architectures for each representative inverter is extracted, as demonstrated with step 5 in Figure 4.10. As explained, the reason for this is to reduce some of the uncertainty of selecting the optimal hyperparameter combination for each inverter within the cluster, based on the preference of the representative inverters.

Table 4.13: FFNN *A* inverter-level Phase-1 best HU and HL combinations.

	1st Choice		2nd Choice	
Inverter	**Hidden Layers**	**Hidden Units**	**Hidden Layers**	**Hidden Units**
I4	2	[128,128]	2	[512,512]
I10	2	[128,128]	3	[64,64,64]
I16	2	[128,128]	3	[64,64,64]
I39	2	[512,512]	3	[64,64,64]
I47	3	[64,64,64]	2	[512,512]
I53	3	[64,64,64]	2	[128,128]
I67	3	[64,64,64]	2	[512,512]
I59	3	[64,64,64]	2	[128,128]
I74	3	[64,64,64]	2	[128,128]
I81	2	[128,128]	3	[64,64,64]

edge case

Table 4.14: FFNN *B* inverter-level Phase-1 best HU and HL combinations.

	1st Choice		2nd Choice	
Inverter	**Hidden Layers**	**Hidden Units**	**Hidden Layers**	**Hidden Units**
I4	2	[128,128]	2	[512,512]
I10	2	[128,128]	2	[1024,1024]
I16	2	[1024,1024]	3	[1024,1024,1024]
I39	2	[128,128]	2	[1024,1024]
I47	2	[128,128]	2	[1024,1024]
I53	3	[128,128,128]	2	[1024,1024]
I67	2	[128,128]	2	[1024,1024]
I59	2	[1024,1024]	2	[128,128]
I74	2	[1024,1024]	3	[1024, 1024, 1024]
I81	2	[128,128]	2	[512,512]

edge case

The hyperparameter selections that are on the edge of the initial grid-search domain are highlighted (blue). In accordance with the guided grid search, these edge case hyperparameter solutions are further extended according to the guided grid-search development strategy (discussed in Section 3.6.2). For example, edge cases for HL=3 and HU=64 are extended so that the HL size becomes HL=4, for which HU sizes of 32, 64 and 128 are explored. Once again, these iterations are executed for both MB sizes of 32 and 64.

Phase-3 development

With the edge cases expanded for Phase-2 of model development, Phase-3 is applied so that models are trained for additional MB sizes of 4, 8, 16, 128 and 256. With all three phases of model development completed, the final HL, HU and MB size selections of the representative inverters are presented in Appendix D in Table D.3 and Table D.4.

Finally, step 11 of the inverter development framework in Figure 4.10 is executed, so that all 84 inverters are trained with the top two identified hyperparameter (HL, HU, MB size) combinations assigned to each cluster. This cluster-based hyperparameter assignment was previously demonstrated with Figure 4.12.

The final forecasting accuracies obtained for these different sets of inverter-level input features are summarised in Table 4.15. Evidently, aggregated inverter-level model *A* delivers the highest overall forecasting accuracy, with a NRMSE value of 8.2 %. Clearly, the addition of WS and WD as low-level input features did not further enhance the inverter-level forecasting accuracy, but instead reduced accuracy. This indicates that the macro-level FFNN

Table 4.15: Validation-set results for aggregated inverter-level FFNN models *A* and *B*.

Forecast model	Input features	NRMSE	MAPE
FFNN *A*	P, GHI, T	**8,20**	**3,50**
FFNN *B*	P, GHI, T, WS, WD	8,39	3,64

model does effectively account for the holistic effects of the low-level variables, which have a non-uniform influence on low-level power output. This observation will be further addressed for the RNN-based inverter-level models, regarding the hypothesis that macro-level model input feature preferences are representative of low-level model input features.

4.4.2.1 Baseline comparison

With the inverter-level models trained, the obtained model forecasting accuracy is once again gauged with a direct comparison to a Linear regression model. As mentioned in Section 4.3.6.1, this baseline model was exhaustively trained with all sliding-window and input feature combinations considered to ensure a fair comparison. The averaged MAPE results are demonstrated in Figure 4.13 for inverters I10, I53 and I67 (top, middle and bottom graphs). These three inverter results are specifically shown due to the practical difficulty of displaying all 84 inverter results simultaneously. Also, these three inverters each represent very different locations within the PV system.

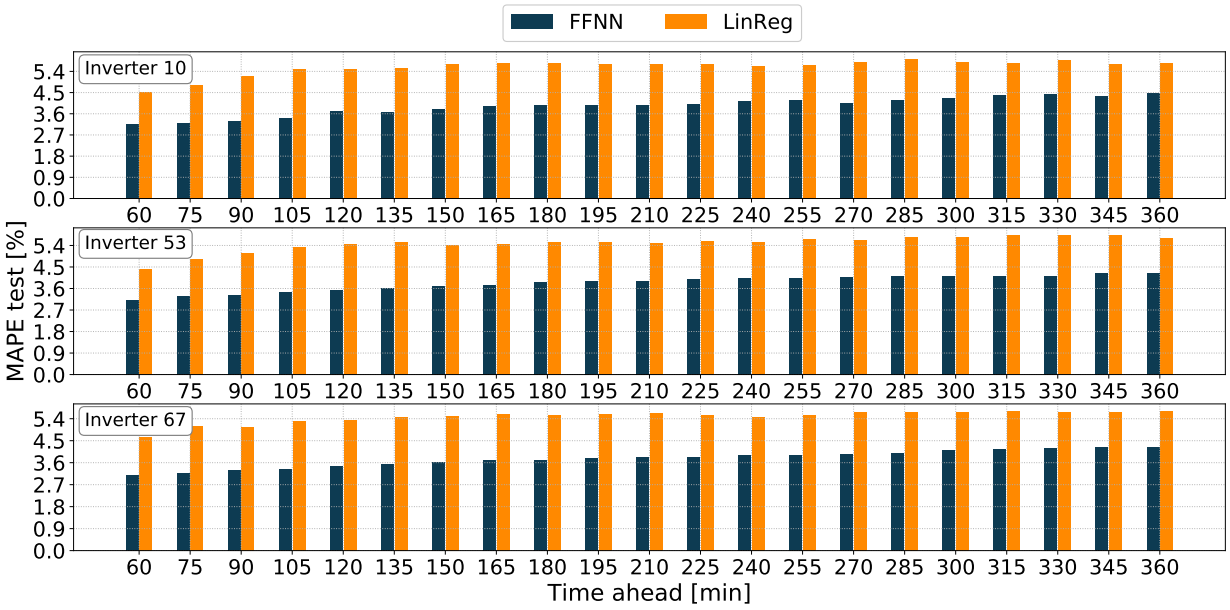


Figure 4.13: Averaged FFNN and Linear regression test set inverter-level MAPE results for all weather conditions as obtained for inverters I10, I53 and I67.

It is evident from Figure 4.13 that the FFNN inverter models are superior to the Linear regression (LinReg) baseline models. Overall, the Linear regression model MAPE averages recorded for inverters I10, I53 and I67 are 5.52 % compared to the FFNN inverter models average of 3.84 %.

4.4.3 FFNN: macro-level vs. inverter-level forecasts

To effectively compare and conclude on the macro-level vs. inverter-level forecasting models, it is necessary to prove that the proposed inverter-clustering approach is indeed an effective strategy to obtain optimal model solutions for all 84 inverters. Therefore, an additional full-stack hyperparameter grid-search is completed for each of the 84 inverters. The execution of this training process has proven to be very intensive and computationally expensive, which allowed for a completely customised parameter and hyperparameter selection to be obtained for each inverter. Overall, a total of 140 h of training time is spent with two NVIDIA Titan Xp GPUs running in parallel, which equates to approximately 30 kWh of energy expenditure. The training time of the macro-level and aggregated inverter-level implementations are also summarised in Table F.1.

With this extensive modelling completed, two sets of aggregated inverter-level forecasts are delivered, together with the macro-level PV-system forecasts. These three different model development approaches are visually demonstrated with Figure 4.14. For the comparison of the results obtained, the forecast results to be presented are labelled and defined as follows:

- FFNN *PV-system* model: PV-system forecasts, as delivered by the macro-level FFNN.
- FFNN *Inverter-cluster* model: Aggregated inverter-level forecasts, with each inverter model trained according to the hyperparameters assigned to the cluster group.
- FFNN *Inverter* model: Aggregated inverter-level predictions, with each inverter model individually trained for an optimal hyperparameter selection.

Figure 4.14: Forecast model approach comparison: PV-system macro-level vs. Inverter-cluster vs. aggregated individual inverter models.

The time-series forecasts of all three FFNN models are presented with Figure 4.15. As previously mentioned, the reason for only illustrating the 1, 3 and 6 h prediction intervals (out of 21 time-steps of 15 min resolution) is due to the overall depiction of forecasting accuracy provided by these intervals. Also, these 6 days (out of 1 year test data set) are specifically shown, because it provides another good variety of all weather conditions.

From the time-series forecasts in Figure 4.15 it is once again evident how forecasting difficulty increases together with forecast horison and the variability of received irradiance. This is

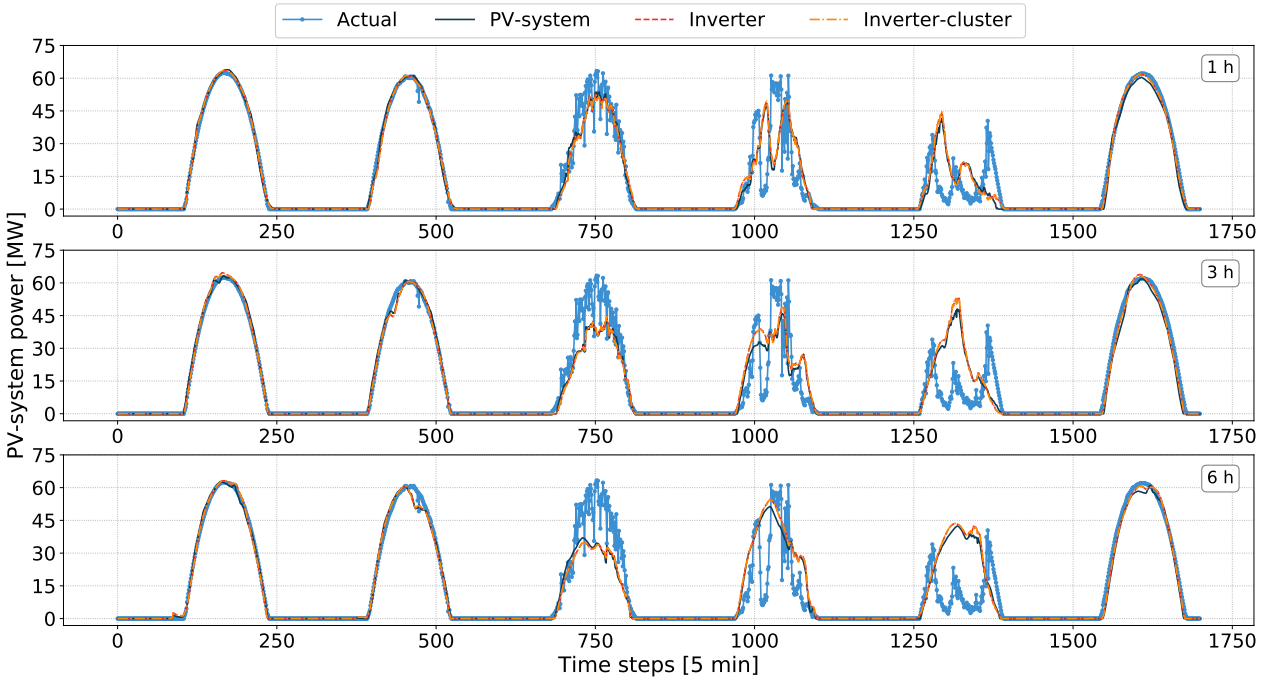


Figure 4.15: Test set time-series forecasts of FFNN PV-system macro-level and aggregated inverter-level models for 1 h, 3 h and 6 h ahead. Day-types from left to right: clear, clear-intermittent, intermittent, intermittent, overcast and clear.

especially evident for the intermittent and overcast day types. From these time-series graphs it is difficult to gauge the model performances, with all three time-series forecasts mostly similar in behaviour. A more exact analysis of the model accuracies is presented with the NRMSE and MAPE forecasting results in Table 4.16.

Table 4.16: Average NRMSE and MAPE test data set results for the macro-level PV-system and aggregated inverter-level FFNN models.

Forecast	**NRMSE [%]**	**MAPE [%]**	**Day-type**
PV-system	3.33	1.78	
Inverter	3.71	2.06	Clear
Inverter-cluster	3.69	2.05	
PV-system	4.86	2.19	Clear-
Inverter	4.80	2.09	Intermittent
Inverter-cluster	4.79	2.08	
PV-system	10.57	4.97	
Inverter	10.53	4.93	Intermittent
Inverter-cluster	10.53	4.93	
PV-system	15.13	8.41	
Inverter	15.21	8.32	Overcast
Inverter-cluster	15.23	8.33	
PV-system	8.19	3.53	
Inverter	**8.15**	**3.45**	All
Inverter-cluster	**8.15**	**3.45**	

The first observation to be made from these averaged NRMSE and MAPE results in Table 4.16 is that the forecasting accuracies of both the aggregated inverter-cluster and individual inverter models are almost exact. This similarity in forecasting ability is given further

context with the NRMSE heatmap plots of Figure 4.16, illustrating all day-types and all prediction points ranging between 1 - 6 h ahead. From these results, the NRMSE heatmap plot is essentially the same for both the FFNN inverter and inverter-cluster forecast models.

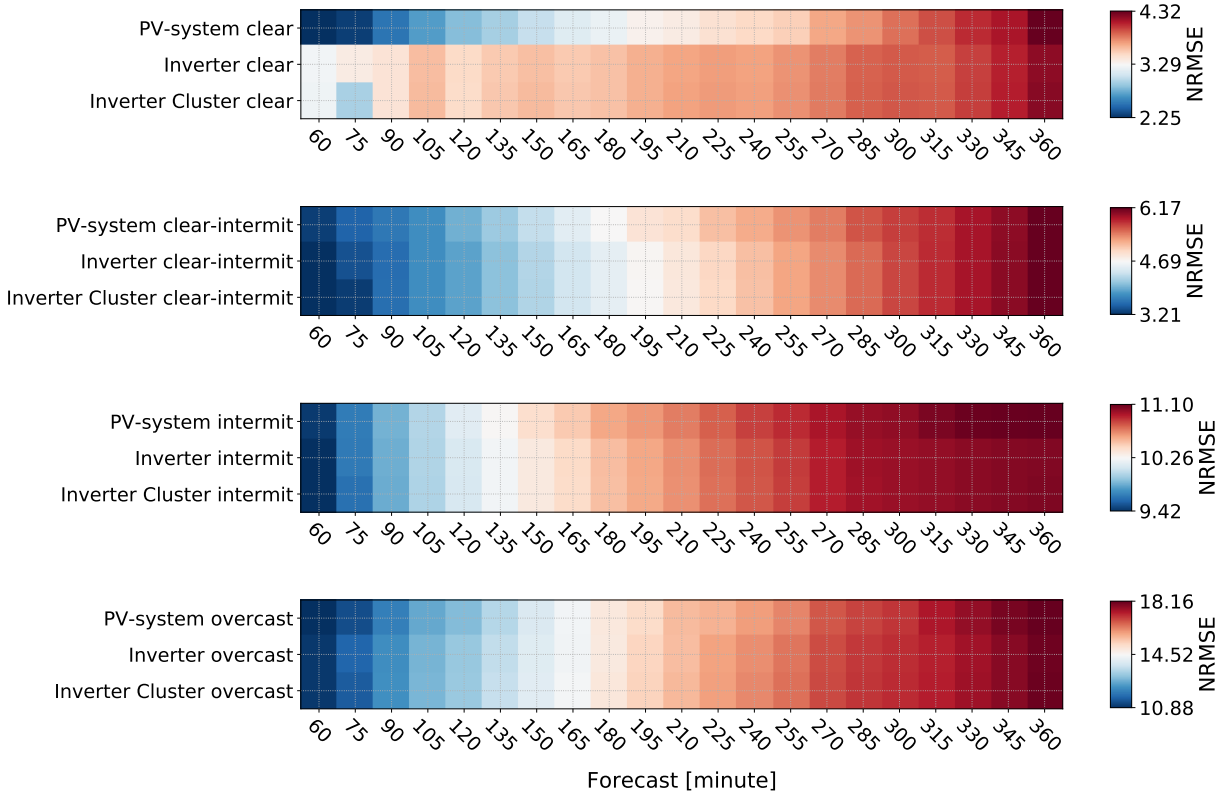


Figure 4.16: Test set NRMSE [%] results of FFNN aggregated inverter-level and macro-level forecast models.

Finally, based on the MAPE Bootstrap CIs, visually demonstrated with Figure 4.17 and summarised in Table 4.17, the ability of the proposed inverter-clustering technique to retrieve highly representative forecasting models for each inverter is evident. Therefore, with the efficacy of the inverter-clustering technique demonstrated, it is shown that a solution has been obtained, which delivers tailored inverter-level forecast models at a reduced computational effort, without sacrificing accuracy. Thus, eliminating the need to execute an extensive hyperparameter grid search for the training of individual inverter models.

Regarding the performance of the aggregated inverter-level forecasts and the PV-system macro-level forecasts, the deterministic results of Table 4.16 indicate that the aggregated inverter-level forecasts deliver a marginal overall performance increase, compared to the macro-level model.

Further interesting to note from the NRMSE and MAPE values, is that the FFNN macro-level forecasts outperform the inverter forecasts on clear days. This observation is given further context with the heatmap plot of Figure 4.16. The macro-level model also marginally outperforms the inverter-level forecasts for overcast days, as seen for the NRMSE values. However, based on the MAPE values, the aggregated inverter models in turn perform slightly better than the macro-level models, indicating that a better overall forecasting accuracy is maintained for overcast day types.

For higher levels of intermittence, the aggregated inverter-level models show an improved forecasting ability, as compared to the macro-level results. It is inferred from these results

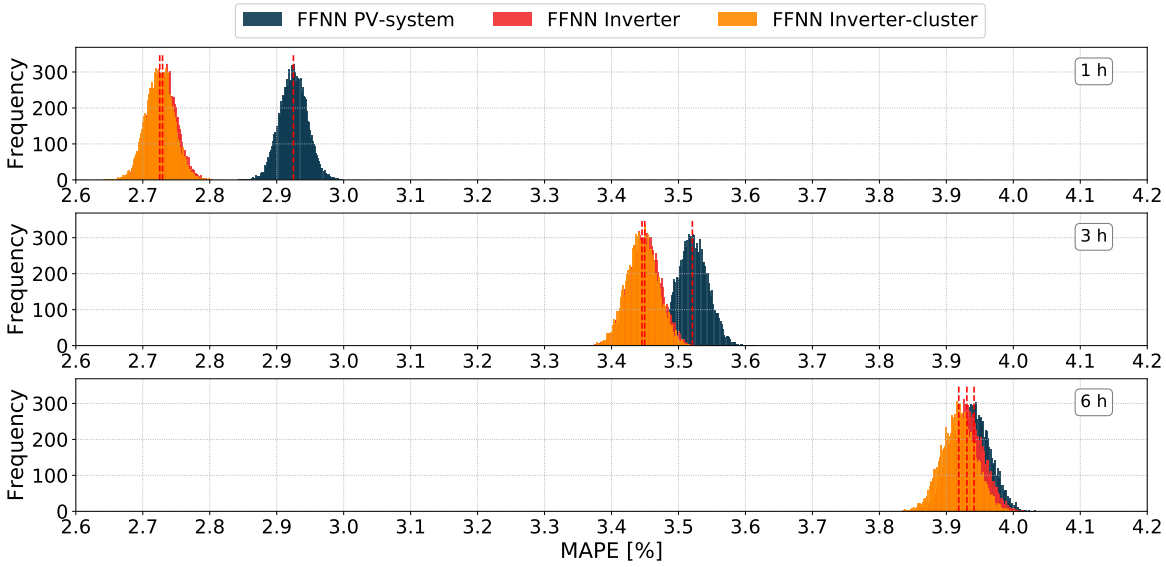


Figure 4.17: FFNN aggregated inverter-level and macro-level PV-system MAPE [%] Bootstrap distributions for all weather types of the test data set.

Table 4.17: The 95 % Bootstrap CIs for all weather conditions for the aggregated inverter and macro-level FFNN models.

Model	Avg. MAPE [%]	Lower Bound [%]	Upper Bound [%]	CI width [kW]
		1 h		
PV-system	2.92	2.89	2.97	59.74
Inverter	2.73	2.69	2.77	59.53
Inverter cluster	2.73	2.69	2.77	59.50
		3 h		
PV-system	3.52	3.47	3.57	68.45
Inverter	3.45	3.40	3.50	68.34
Inverter cluster	3.45	3.40	3.49	68.41
		6 h		
PV-system	3.94	3.89	3.99	74.48
Inverter	3.93	3.88	3.98	74.81
Inverter cluster	3.92	3.87	3.97	74.80

that there is a more homogenous exposure of the overall PV-system to the environment (with particular reference to GHI), on clear and overcast days. This reduced variability allows the FFNN macro-level PV-system forecast model to sufficiently emulate the low-level PV-system dynamics as well. Days with higher intermittence result in higher levels of non-homogenous PV-system exposure to the environment and in particular GHI, which in most cases can result in partial PV-system shading. From these results it is further concluded that the FFNN inverter-level models are inherently over-sensitive to the dynamics of the surrounding environment and the PV system. Thus, based on the results, it is observed that the aggregated inverter-level models are more capable of capturing this dissimilarity of low-level power output dynamics. Therefore, more accurate forecasts are delivered for these intermittent day-type scenarios.

From the more interpretable MAPE Bootstrap CI analysis presented with Figure 4.17 and summarised in Table 4.17, it is clear that with an increase in forecast horison, there is a decrease in the difference between the centre lines of the macro-level and inverter-level MAPE distributions. These results convey that the performance increase obtained with the

aggregated inverter-level forecasts is primarily relevant to shorter forecast horisons. This observed short-term advantage can be attributed to the low-level effects of the surrounding environment on the PV-system, which are more predictable for the inverter models than the macro-level model. However, as forecast time extends and it becomes more difficult to anticipate the environmental and atmospheric conditions, so does the gained advantage from FFNN inverter-level forecasts diminish.

4.4.4 GRU-RNN model development

Based on the macro-level forecasting results as provided in Section 4.3, the LSTM-RNN model is outperformed by the GRU-RNN model. Therefore, with the GRU-RNN model identified as the most accurate RNN-based model, the research regarding low-level forecasting is continued with this model.

Once again, the same three-phase development process, demonstrated with Figure 4.10, is applied for the training of the GRU-RNN inverter-level models. Also, given the efficacy of the inverter-clustering technique (demonstrated in Section 4.4.3) it is once again applied for training the individual inverter models.

From the macro-level GRU-RNN model results obtained, the P, GHI, T and AP input features delivered the highest accuracy. Thus, the addition of AP as high-level input feature will also be investigated. The use of RH is not considered, since this high-level input feature did not improve the macro-level GRU-RNN forecasting ability (Table 4.4). However, since WS and WD are regarded as low-level input features, the effects of these input features will also be determined. Importantly, these input feature combinations are to be progressively evaluated with model comparisons labelled as follows:

- *C*: Inverter-level GRU-RNN model with input features: P, GHI, T
- *D*: Inverter-level GRU-RNN model with input features: P, GHI, T, AP
- *E*: Inverter-level GRU-RNN model with input features: P, GHI, T, (AP)[1], WS, WD

Phase-1 development

The same initial Phase-1 hyperparameter solution space is used, which was assigned for the GRU-RNN macro-level model development, as summarised in Table 4.18. Once again, important to note from Table 4.18 is the sliding-window size of 6 h. Similar to the FFNN model development, unnecessary computational effort is avoided with a pre-defined sliding-window size for the given sets of input features.

Table 4.18: GRU-RNN inverter-level model development Phase-1 solution space.

Model	HLs	HUs per HL	MB sizes	Sliding window	Input features
GRU-RNN *C*	2,3	16, 32, 64, 128, 256, 512	32, 64	6 h	P, GHI, T
GRU-RNN *D*					P, GHI, T, AP
GRU-RNN *E*					P, GHI, T, (AP), WS, WD

[1]This input feature might be included or not, dependent on the results obtained for GRU-RNN aggregated inverter-level models *C* and *D*

However, this pre-defined sliding-window size is selected after evaluating the 1, 2, 3, 6 and 24 h sliding windows, for the hyperparameters as defined in Table 4.18. The results obtained for the given set of input features of models *C*, *D* and *E* are shown in Appendix D.3 in Tables D.5 - D.8. From these results, a sliding-window size of 6 h delivered the highest accuracies for the sample-set of three representative inverters (I10, I52, I67), each strategically selected due to the variation of location within the PV-system. Once again, similar to the FFNN models, the macro-level and inverter-level models have shown a preference for the same sliding-window size.

Phase-2 development

The architectures identified for each of the representative inverters, which deliver the highest forecasting accuracies for Phase-1 of model development are presented in Table D.9 and Table D.10 in Appendix D.4. With the top two architectures extracted for each representative inverter, several edge cases are identified for the initial grid-search space. These edge case hyperparameter solutions are further extended according to the guided grid-search development strategy (discussed in Section 3.6.2).

Phase-3 development

With Phase-1 and Phase-2 completed for inverter-level models *C* and *D*, the identified model architectures are trained for the additional MB sizes of 4, 8, 16, 128 and 256. From these iterations the final GRU-RNN inverter models, as optimised for each of the representative inverters, are summarised in Table D.13 and Table D.14 in Appendix D.4. Finally, with models trained for all 84 inverters, the aggregated inverter-level forecasting results are summarised in Table 4.19.

From these NRMSE results of Table 4.19, the addition of AP as input feature delivers a very similar forecasting ability compared to the use of only P, GHI and T as historic input features, with a NRMSE difference of 0.01 %. A similar scenario was observed during the development of the macro-level models, as previously summarised in Table 4.4, where both of these input feature sets also delivered very similar NRMSE results.

Due to the difficulty of making an absolute decision regarding the ideal set of input features, it is decided to extend both models *C* and *D* to include WS and WD. Therefore, to evaluate the potential of WS and WD to improve inverter-level forecasts, the next GRU-RNN inverter-level model input feature set is defined and labelled as:

- *E*: Aggregated inverter-level GRU-RNN with input features: P, GHI, T, WS, WD
- *F*: Aggregated inverter-level GRU-RNN with input features: P, GHI, T, WS, WD, AP

Once again, the model development is executed according to the defined framework of Figure 3.19, so that the best aggregated inverter-level forecasting models are obtained for both sets of these input features. Firstly, a pre-defined sliding-window size is determined from the

Table 4.19: Comparison of NRMSE and MAPE validation-set results for aggregated inverter-level GRU-RNN models C and D.

Forecast model	Input features	NRMSE	MAPE
GRU-RNN *C*	P, GHI, T	**8.077**	3.447
GRU-RNN *D*	P, GHI, T, AP	8.087	**3.439**

sample-set of representative inverters I10, I53 and I67. From the results obtained, presented in Appendix D.3 in Table D.7 and Table D.8, the preferred sliding-window size is once again established as 6 h for both models *E* and *F*.

With the sliding-window size defined, model development can commence so that Phase-1, Phase-2 and Phase-3 is executed. Finally, the model hyperparameters obtained, which deliver the best forecasting accuracy for each representative inverter are displayed in Appendix D.5 in Table D.15 and Table D.16. With all 84 inverters trained accordingly, the highest validation set forecasting accuracies obtained are summarised in Table 4.20.

Table 4.20: Comparison of NRMSE and MAPE validation-set results for aggregated inverter-level GRU-RNN models E and F.

Forecast model	Input features	NRMSE	MAPE
Inverter-cluster C	P, GHI, T	**8.077**	3.447
Inverter-cluster D	P, GHI, T, AP	8.087	**3.439**
Inverter-cluster E	P, GHI, T, WS, WD	8.126	3.505
Inverter-cluster F	P, GHI, T, WS, WD, AP	8.127	3.496

Evidently, the addition of WS and WD as input features, reduces the forecasting ability of both models *C* and *D*. Therefore, the analysis is continued with the selection of aggregated inverter-level model *D*. Apart from the small difference in the NRMSE value compared to model *C*, by selecting model *D* it allows the analysis to be more analogous to the macro-level GRU-RNN model, which delivered the best results with P, GHI, T and AP as input features. This further compliments the direct comparison of the GRU-RNN inverter-level and macro-level models, considering that exactly the same input feature data set is used for both.

4.4.4.1 Baseline comparison

As a final measure to gauge the obtained inverter-level GRU-RNN model results, a baseline comparison is made to a Linear regression model in Figure 4.18

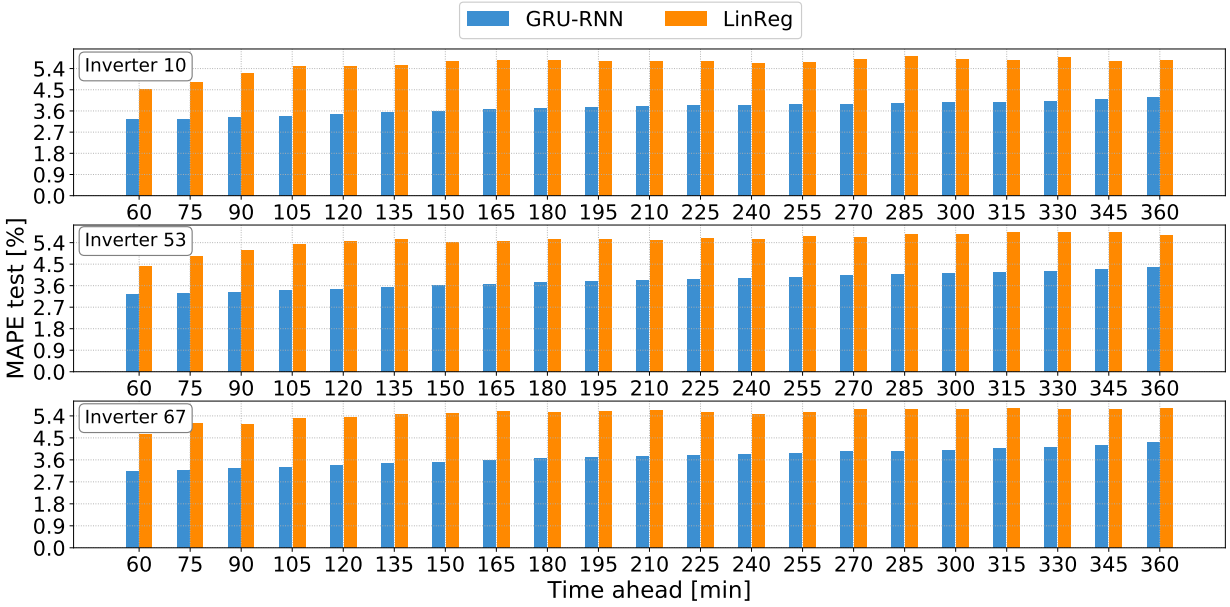


Figure 4.18: Averaged GRU-RNN and Linear regression test set inverter-level MAPE results for all weather conditions as obtained for inverters I10, I53 and I67.

As mentioned in Section 4.3.6.1 and Section 4.4.2.1, this baseline model was exhaustively trained to ensure a fair comparison. The averaged MAPE results are once again demonstrated for inverters I10, I53 and I67 (top, middle and bottom graphs) in Figure 4.18, which represent very different locations within the PV system.

Similar to the FFNN results, it is evident from Figure 4.18 that the GRU-RNN inverter models are superior to the Linear regression models. Overall, the LinReg model MAPE averages recorded for inverters I10, I53 and I67 are 5.52 % compared to the FFNN inverter models average of 3.75 %.

4.4.5 GRU-RNN: macro-level vs. inverter-level forecasts

With the exhaustive hyperparameter optimisation and training procedures completed, the time-series forecasting ability of the macro-level and inverter-level GRU-RNN models are demonstrated with Figure 4.19. From this test data segment, various and difficult weather types are on display, with a complete summary of overall model performances provided in Table 4.21.

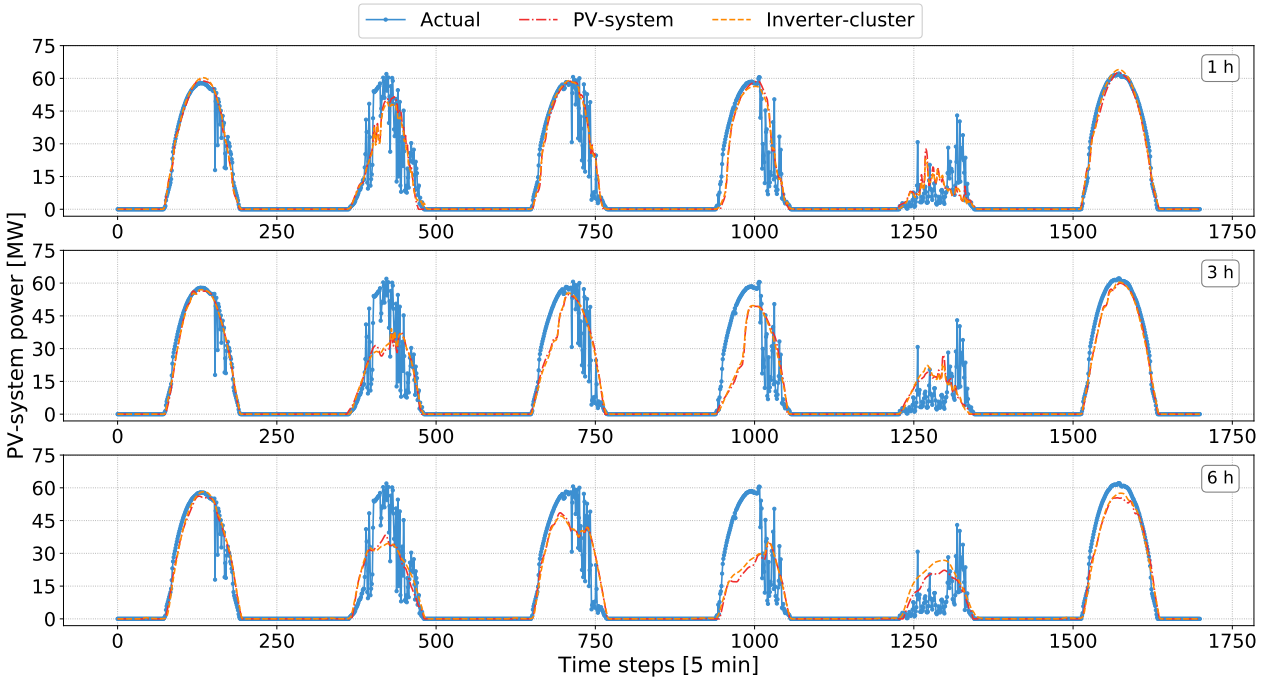


Figure 4.19: Test set time-series forecasts of GRU-RNN PV-system macro-level and aggregated inverter-level models for 1, 3 and 6 h ahead. Day-types from left to right: clear-intermittent, intermittent, intermittent, intermittent, overcast and clear.

Table 4.21: Test set NRMSE and MAPE results for the macro-level and aggregated inverter-level GRU-RNN models.

Forecast	NRMSE [%]	MAPE [%]	Day-type
PV-system	2.67	1.42	Clear
Inverter-cluster	2.45	1.27	
PV-system	4.75	2.09	Clear-
Inverter-cluster	4.68	2.03	Intermittent
PV-system	10.53	4.92	Intermittent
Inverter-cluster	10.39	4.88	
PV-system	14.76	8.13	Overcast
Inverter-cluster	14.99	8.29	
PV-system	8.12	3.42	All
Inverter-cluster	**8.02**	**3.39**	

Contrary to the aggregated FFNN inverter-level forecasts, the macro-level PV-system forecasts are outperformed by the GRU-RNN inverter-level forecasts for clear days. However, similar to the results of the FFNN model, the GRU-RNN macro-level PV-system forecast model delivers better overcast day accuracies.

More context regarding the NRMSE accuracies for each prediction interval and day type is given in the heatmap plot of Figures 4.20. From this overall prediction accuracy depicted by the heatmap, it is once again evident that the aggregated inverter-level models outperform the macro-level model for the clear, clear-intermittent and intermittent day types. Further to note is the ability of the aggregated inverter-level models to maintain a good prediction accuracy beyond 5 h ahead for clear days, as compared to the deviating accuracy of the macro-level model for this forecast scenario.

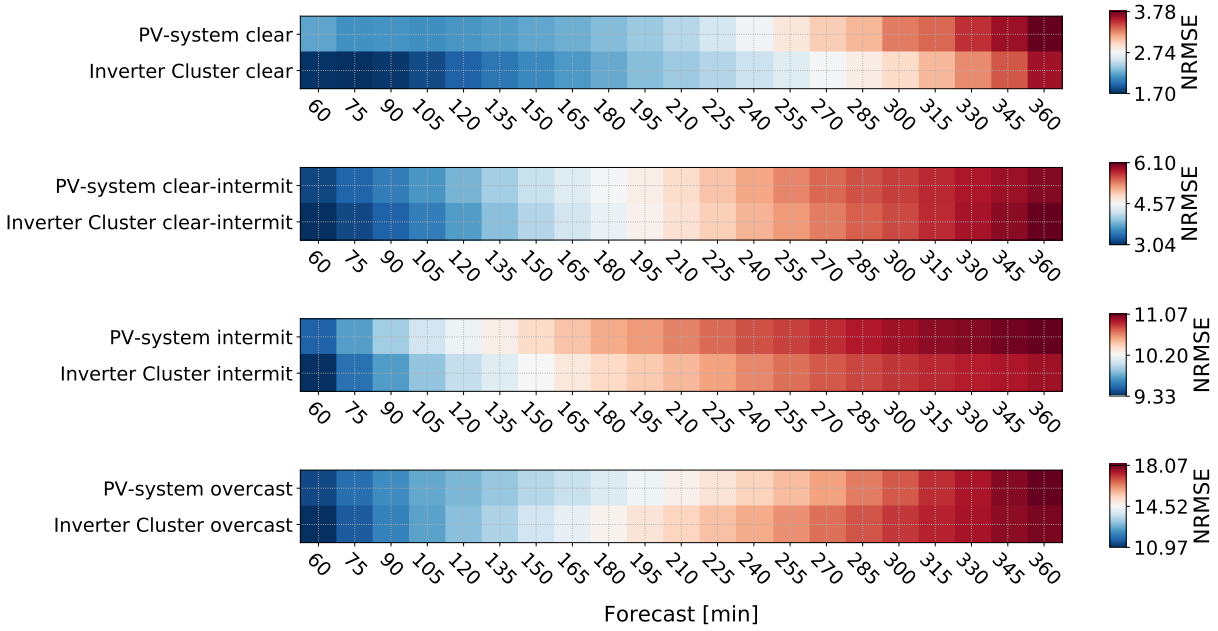


Figure 4.20: Test set NRMSE [%] results of GRU-RNN aggregated inverter-level and macro-level forecast models.

The more interpretable MAPE Bootstrap CIs are presented with Figure 4.21 and are further summarised in Table 4.22. Another aspect to note regarding the results presented in Table 4.22, is that the CI widths are very similar for both aggregated inverter-level and macro-level models. This indicates that the certainty of the delivered forecasts, regarding the 95 % CI distributions of the MAPE values is very similar. This shows that the performance increase seen for the aggregated inverter-level forecasts is primarily with regards to the deterministic error, but there is no noteworthy performance increase in terms of the CI certainty distributions of the MAPE values.

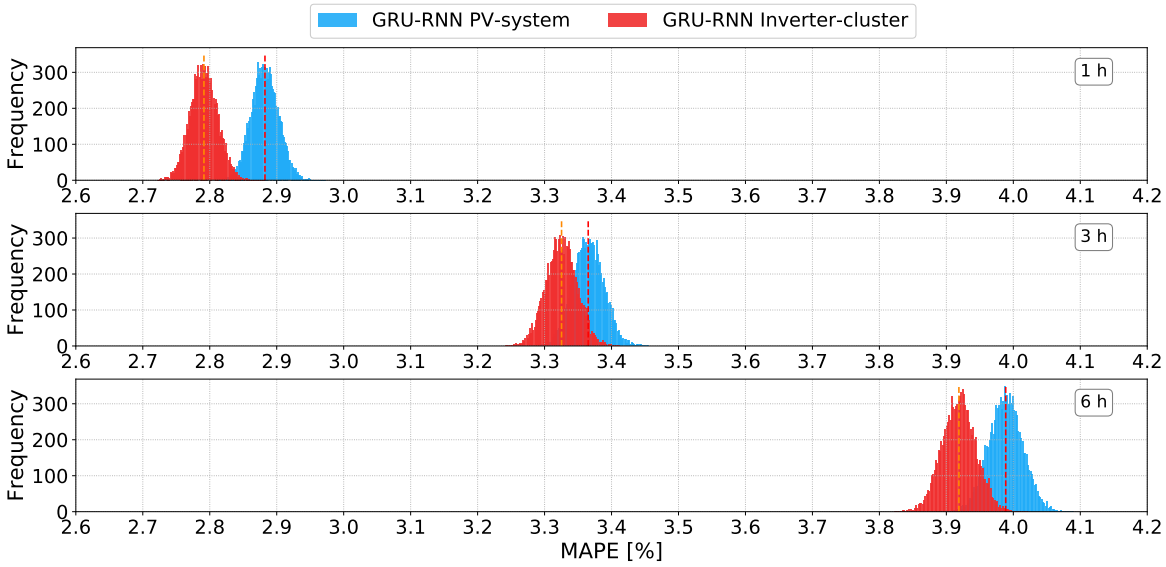


Figure 4.21: GRU-RNN inverter-level and macro-level PV-system MAPE [%] Bootstrap distributions for all weather types.

Table 4.22: The 95 % Bootstrap CIs for all weather conditions for the aggregated inverter and macro-level GRU-RNN models.

Model	Avg. MAPE [%]	Lower Bound [%]	Upper Bound [%]	CI width [kW]
		1 h		
PV-system	2.88	2.84	2.92	59.82
Inverter cluster	2.79	2.75	2.83	59.70
		3 h		
PV-system	3.37	3.32	3.41	67.33
Inverter cluster	3.33	3.28	3.37	67.72
		6 h		
PV-system	3.99	3.94	4.04	74.43
Inverter cluster	3.92	3.87	3.97	73.71

4.5 Summary of results

4.5.1 Direct comparison of FFNN and GRU-RNN models

Considering that the inverter-level and macro-level models are completed, the final question that remains is regarding the direct comparison of the FFNN and GRU-RNN models. The

heatmaps in Figure 4.22 provides perspective on the deterministic MAPE results obtained for all day types, as delivered for each 15 min time step of the 1 - 6 h ahead forecast interval.

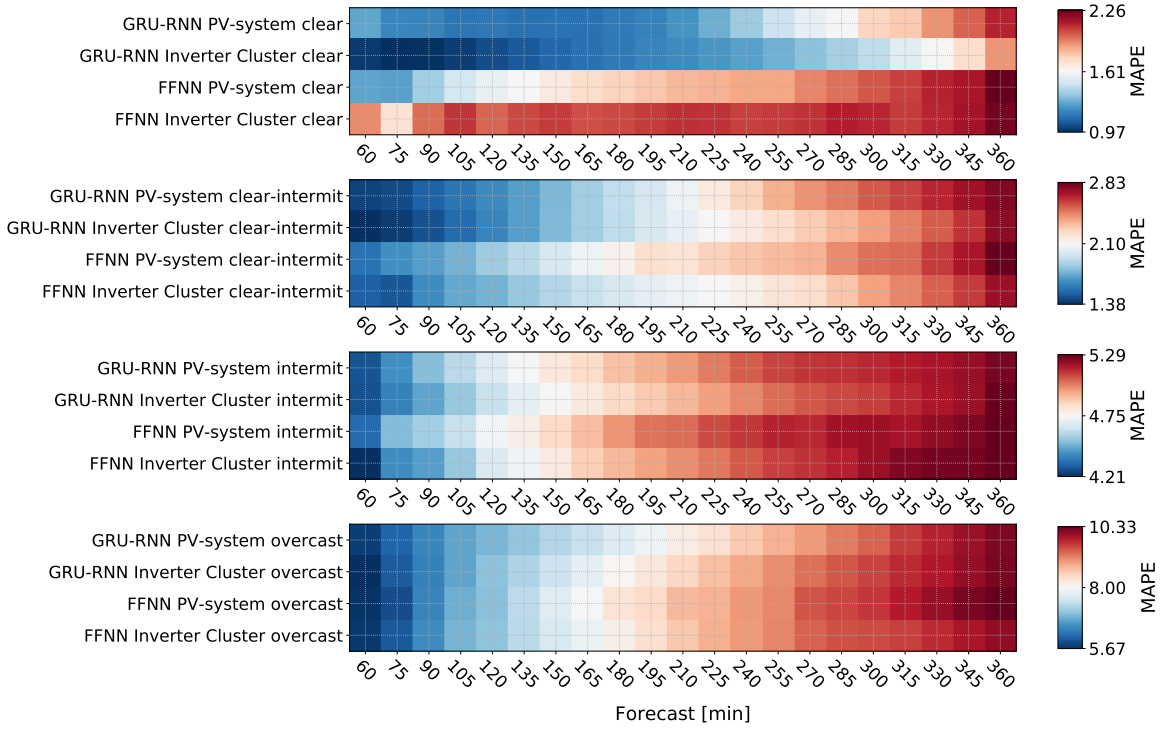


Figure 4.22: Inverter-level and macro-level test set MAPE [%] results for GRU-RNN and FFNN models.

With reference to the heatmap summary in Figure 4.22, this direct comparison of the GRU-RNN and FFNN forecast models indicates that the GRU-RNN model has delivered the most accurate forecasting accuracies for both the macro-level and aggregated inverter-level predictions. This is further evident from the FFNN and the GRU-RNN results summarised in Table 4.23.

Table 4.23: Summary of aggregated inverter-level and macro-level PV-system NRMSE and MAPE results

Forecast model	**NRMSE [%]**		**MAPE [%]**		**Day-type**
	FFNN	**GRU-RNN**	**FFNN**	**GRU-RNN**	
PV-system	3.33	2.67	1.78	1.42	Clear
Inverter-cluster	3.69	**2.45**	2.05	**1.27**	
PV-system	4.86	4.75	2.19	2.09	Clear-
Inverter-cluster	4.79	**4.68**	2.08	**2.03**	Intermittent
PV-system	10.57	10.53	4.97	4.92	Intermittent
Inverter-cluster	10.53	**10.39**	4.93	**4.88**	
PV-system	15.13	**14.76**	8.41	**8.13**	Overcast
Inverter-cluster	15.23	14.99	8.33	8.29	
PV-system	8.19	8.12	3.53	3.42	All
Inverter-cluster	8.15	**8.02**	3.45	**3.39**	

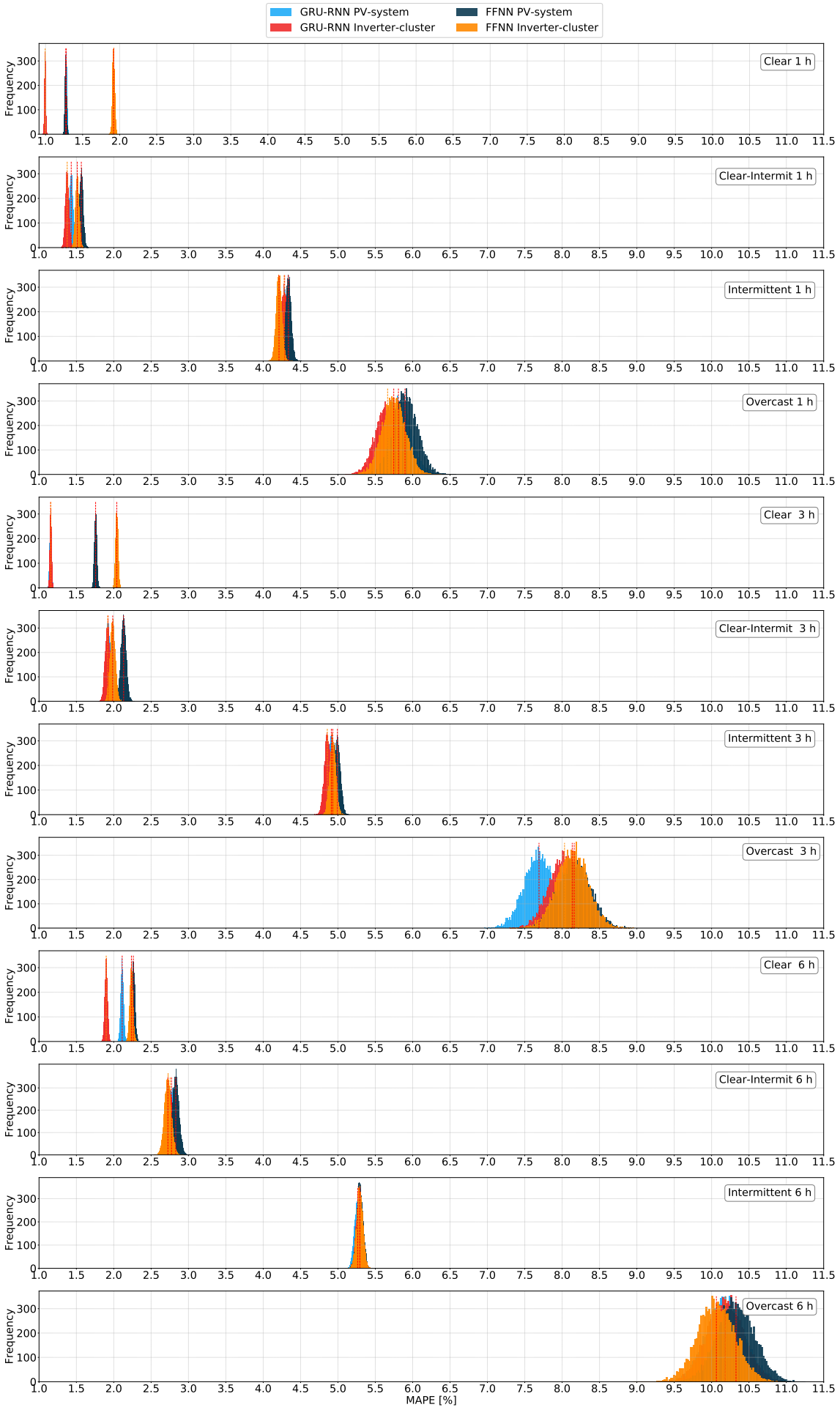


Figure 4.23: All test set MAPE Bootstrap CIs for the GRU-RNN and FFNN aggregated inverter-level and macro-level forecast accuracies.

For a visual observation of the uncertainty associated with these forecast models, Figure 4.23 demonstrates the MAPE 95 % Bootstrap CI graphs of the macro-level and inverter-level models. From these MAPE heatmap results and the 95 % Bootstrap CI distributions, the conclusions made regarding an overall comparison of the FFNN and GRU-RNN, as applied for the inverter-level and macro-level forecasting methodologies are:

- The overall difference between the FFNN compared to the GRU-RNN model accuracies becomes less as the the forecast horison increases from 1 h to 6 h ahead, as evident from Figure 4.23. This is more prevalent regarding the intermittence of the received irradiance, with a high intermittence resulting in less of a difference between the GRU-RNN and FFNN model accuracies. This relationship is displayed with Figure 4.24.
- From the averaged MAPE forecast results, as determined for the clear and clear-intermittent days, the GRU-RNN macro-level model has shown an improved ability to capture the inverter-level power output dynamics. Therefore, outperforming the aggregated inverter-level FFNN accuracy.
- Regarding the uncertainty of the delivered forecasts, the major contributor towards forecast uncertainty is evidently the increased variability in received irradiance, which is further amplified by an increased forecast horison, as evident from the presented Bootstrap CI distributions. Also, it is observed that the inverter-level forecasts made no considerable difference in the Bootstrap uncertainty captured by the CI widths. Therefore, although the inverter-level forecasts provide a marginal increase in forecasting accuracy, there is no notable difference in the uncertainty distributions as seen with the CIs.

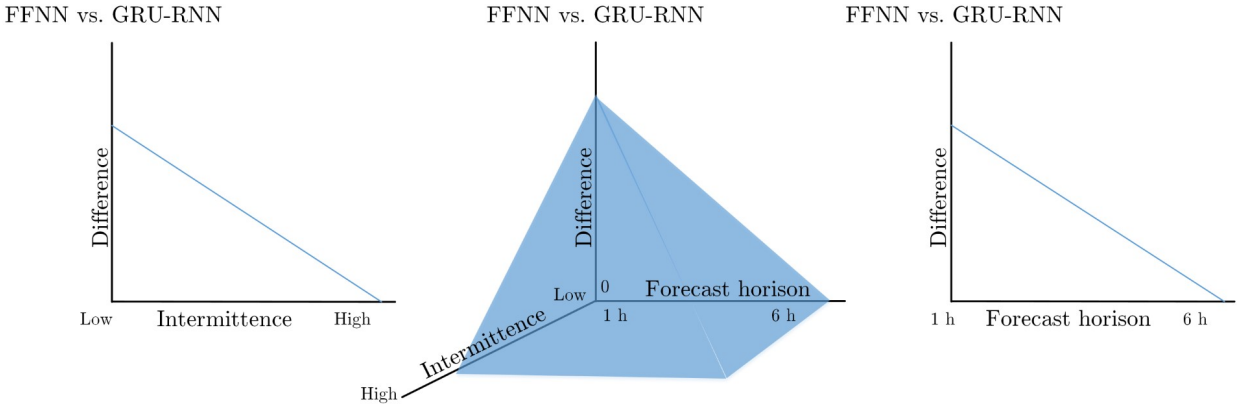


Figure 4.24: Relationship of FFNN and GRU-RNN forecasting accuracy differences.

Regarding the process of model training, it was observed that the proposed aggregated inverter-level forecast models are more robust to bad data segments than the macro-level models. Considering the availability of data for all 84 inverters, it was possible to successfully substitute bad data segments with data from other inverters, without affecting inverter-level model training. Another advantage of the low-level forecasts is that a bad inverter forecast, due to low data quality, only makes a small contribution towards the total predicted PV-system power output. However, since the macro-level models are trained with a single set of power data, as measured at the point of grid connection, there is a higher model sensitivity and risk towards bad data segments. For these reasons, the proposed low-level forecasting methodology has shown to be more robust towards bad data quality. Finally, the total macro-level and aggregated inverter-level model training times are presented in Appendix F.

4.5.2 Real-world forecasting results

Considering that the inverter-level and macro-level models are completed, the final question that remains is regarding how these models performed for an applied real-world forecasting scenario? Up to this point, the results obtained have indeed been representative of the real-world forecasting ability of these models. However, as also mentioned by T. Hoff *et al.* [34], the use of day-time-only forecasting accuracy serves as a practical real-world quantifier for decision makers (PV-system operators, grid managers, etc.).

The reason being due to the calculation of the MAPE and NRMSE (Eq. (4.1) and (4.2)) values, which are determined based on the average of all the prediction points. With night-time forecasts considered in the calculation, this does reduce the true day-time forecasting error proportionally, since night-time forecasts are very predictable and forecasts essentially have a 0 % error for this time period.

Importantly, this does not rule out the previous results obtained, because models had to be compared for a full 24 h forecast scenario, to ensure that forecasts made up to 6 h in advance for the early morning and late afternoon hours are accurate. Also, with the forecast values normalised (NRMSE, MAPE) to PV-system capacity, the errors made for the day-time predictions are automatically prioritised. Therefore, the initial result sections were aimed at establishing the overall differences in forecasting ability between the inverter-level and macro-level models. With this goal accomplished, the last step towards completing the results section is the analysis of day-time-only forecasts.

With this considered, the day-time-only accuracies obtained are summarised for the hourly forecasts in Table 4.24. The MAPE difference between the aggregated inverter-level and macro-level errors is determined as defined by Eq. (4.6). Therefore, a positive difference indicates a better forecasting ability by the inverter-level model (smaller error). Also, the percentage improved (positive value) forecasting accuracy of the inverter-level models relative to the PV-system macro-level models is obtained with Eq. (4.7).

Table 4.24: Day-time-only FFNN and GRU-RNN model test set MAPE results with a positive difference indicating an improved forecasting accuracy by the inverter-level models.

			Time steps					
Weather Type	**Model**	**Forecast**	**60**	**120**	**180**	**240**	**300**	**360**
Clear	GRU-RNN	PV-system	2.52	2.27	2.29	2.74	3.47	4.19
		Inverter-level	1.95	2.06	2.29	2.51	2.87	3.73
		Difference	0.57	0.21	0.00	0.24	0.60	0.46
		% Improvement	22.74	9.41	0.11	8.65	17.25	10.87
	FFNN	PV-system	2.49	3.12	3.49	3.70	4.01	4.47
		Inverter-level	3.79	3.95	4.06	4.08	4.19	4.40
		Difference	-1.30	-0.84	-0.56	-0.38	-0.18	0.07
		% Improvement	-52.03	-26.84	-16.15	-10.34	-4.56	1.56
Clear-Intermittent	GRU-RNN	PV-system	3.01	3.46	4.08	4.81	5.41	5.89
		Inverter-level	2.88	3.42	4.06	4.60	5.09	5.81
		Difference	0.13	0.04	0.02	0.21	0.32	0.07
		% Improvement	4.16	1.04	0.44	4.39	5.93	1.27
	FFNN	PV-system	3.28	3.91	4.51	4.88	5.32	5.96
		Inverter-level	3.17	3.72	4.21	4.54	5.09	5.75
		Difference	0.11	0.19	0.31	0.34	0.23	0.21
		% Improvement	3.22	4.83	6.79	7.01	4.38	3.53
Intermittent	GRU-RNN	PV-system	8.86	9.67	10.14	10.49	10.66	10.85
		Inverter-level	8.83	9.56	10.02	10.35	10.59	10.87
		Difference	0.04	0.11	0.12	0.14	0.07	-0.02
		% Improvement	0.41	1.09	1.19	1.37	0.64	-0.22
	FFNN	PV-system	8.97	9.79	10.30	10.60	10.76	10.90
		Inverter-level	8.71	9.65	10.18	10.49	10.78	10.90
		Difference	0.26	0.13	0.12	0.10	-0.01	0.00
		% Improvement	2.89	1.38	1.18	0.95	-0.13	0.00
Overcast	GRU-RNN	PV-system	12.35	14.84	16.56	18.65	20.40	21.97
		Inverter-level	11.97	15.09	17.33	19.31	20.93	22.13
		Difference	0.38	-0.24	-0.77	-0.67	-0.54	-0.15
		% Improvement	3.05	-1.64	-4.64	-3.58	-2.64	-0.70
	FFNN	PV-system	12.09	14.94	17.73	19.38	21.01	22.61
		Inverter-level	12.21	15.09	17.50	19.44	20.75	22.00
		Difference	-0.12	-0.16	0.23	-0.05	0.26	0.61
		% Improvement	-0.98	-1.04	1.32	-0.27	1.24	2.69
All	GRU-RNN	PV-system	5.77	6.42	6.91	7.37	7.66	7.98
		Inverter-level	5.75	6.33	6.85	7.26	7.58	8.02
		Difference	0.03	0.08	0.06	0.11	0.08	-0.04
		% Improvement	0.44	1.27	0.83	1.52	0.98	-0.45
	FFNN	PV-system	6.03	6.72	7.26	7.60	7.82	8.08
		Inverter-level	5.63	6.50	7.11	7.45	7.79	8.04
		Difference	0.40	0.23	0.15	0.15	0.03	0.04
		% Improvement	6.70	3.39	2.03	1.96	0.45	0.45

$$\text{MAPE Difference} = \text{MAPE PV-system} - \text{MAPE Inverter-level} \tag{4.6}$$

$$\%\ \text{Improvement} = \frac{\text{MAPE Difference}}{\text{MAPE PV-system}} \times 100\% \tag{4.7}$$

From these results, the highest forecasting improvements are observed for the inverter-level GRU-RNN model for clear-day forecasts, with the hourly forecast improvements ranging between 0 % to 23 %. Regarding the FFNN inverter-level model, the highest relative increase in forecasting accuracy (compared to FFNN macro-level model) is observed for clear-intermittent days, with values ranging between 3 % to 7 % for the hourly forecasts. For the inverter-level forecast models the highest reduction in forecasting accuracy, relative to the macro-level forecasts, is seen for the FFNN clear-day forecasts, with values ranging between -4.5 % to -52 %.

For a more detailed overview of the day-time-only comparisons of inverter-level and macro-level forecasts, Figure 4.25 illustrates the difference in the MAPE values for all forecast time steps.

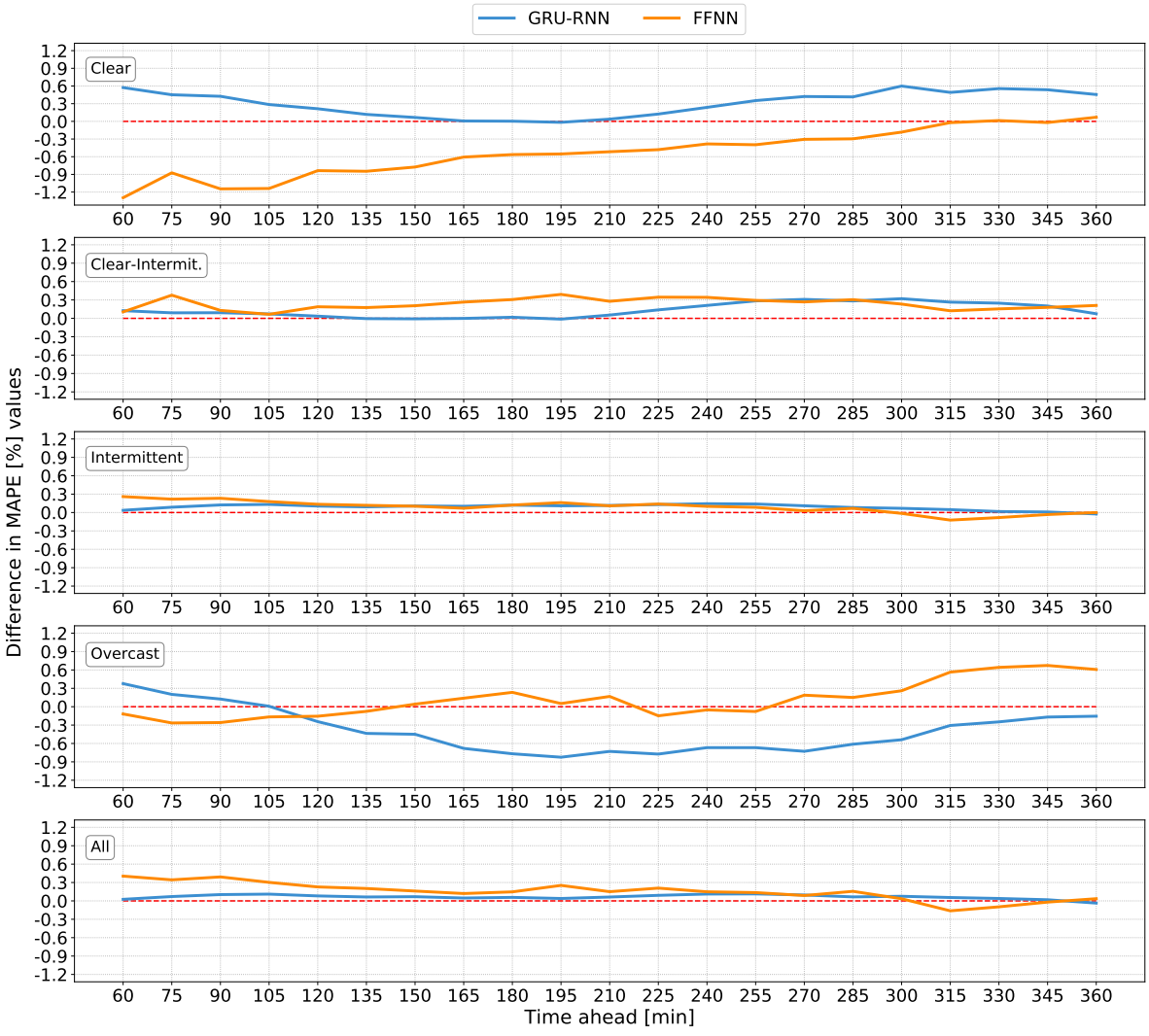


Figure 4.25: Difference between the inverter-level and macro-level MAPE results obtained, with positive difference values indicating the improved forecasting accuracy from the inverter-level models.

Once again, from the graphs in Figure 4.25, a positive difference value above the red dash-line indicates an improved forecasting accuracy by the aggregated inverter-level models. Immediately evident from these graphs is that the true value of the inverter-level forecasting models is seen for the clear-intermittent and intermittent day types. This is especially

evident for the % improvement values recorded for the clear-intermittent day type values in Table 4.24.

Given the deterministic results in Table 4.24, the 95 % Bootstrap CIs obtained for all weather conditions are summarised in Table 4.25. More specific details regarding the 95 % Bootstrap CIs obtained are presented in Table E.1 - E.4 in Appendix E, with a visual demonstration of the CI distributions for all day types presented with Figure E.1.

Table 4.25: Day-time-only 95 % Bootstrap CIs obtained for all weather conditions for the aggregated inverter-level and macro-level PV-system FFNN and GRU-RNN models.

Forecast model		Avg. MAPE [%]	Lower Bound [%]	Upper Bound [%]	CI width [kW]
			1 h		
GRU-RNN	PV-system	5.77	5.70	5.85	111.67
	Inverter cluster	5.75	5.68	5.82	109.50
FFNN	PV-system	6.03	5.96	6.11	108.52
	Inverter cluster	5.63	5.55	5.70	111.31
			3 h		
GRU-RNN	PV-system	6.91	6.83	6.99	125.98
	Inverter cluster	6.85	6.77	6.94	124.18
FFNN	PV-system	7.26	7.18	7.35	124.72
	Inverter cluster	7.11	7.03	7.20	123.53
			6 h		
GRU-RNN	PV-system	7.99	7.90	8.07	132.90
	Inverter cluster	8.02	7.94	8.11	130.81
FFNN	PV-system	8.08	7.99	8.17	133.93
	Inverter cluster	8.04	7.95	8.13	132.82

Ultimately, the most accurate overall forecasting performances are obtained with the aggregated inverter-level GRU-RNN model. The macro-level GRU-RNN model also shows a superior ability to capture low-level PV-system behaviour, as compared to the multi-layer FFNN macro-level model.

Chapter 5

Conclusions and recommendations

5.1 Research summary

To conclude on the research, the following has been presented:

Chapter 1

Given the characteristic variability of solar PV systems as energy source, the need for accurate power forecasting was firstly clarified within the context of commercial power production and improved grid stability. This was followed with a brief synopsis regarding the fundamentals of PV forecasting. With the fundamentals explained, a holistic overview of published forecasting solutions was given, with a specific focus on two segments of more recent PV-forecasting literature, namely: Classic ML model solutions and State-of-the-art DL model solutions.

After presenting the in-depth analysis of published literature, a clear knowledge gap was identified. This knowledge gap being that forecast models are traditionally trained with a macro-level forecasting approach. It was further identified that the scarcity of forecasting research applied to large multi-megawatt PV systems only gave further rise to the disconnect between available research and real-world, commercial PV power production.

In an important attempt towards improved utility-scale PV forecasting accuracy, a set of research objectives were formulated based on two questions:

1 *Is it possible to obtain an enhanced forecast model sensitivity to non-uniform power output dynamics, by using an aggregate of low-level forecasts?*

2 *With state-of-the-art deep learning models trained as a macro-level forecasting solution, what is the ability of these more powerful models to capture low-level, non-uniform PV-system power output dynamics?*

Chapter 2: Data processing

With the proposed forecasting methodology applied to a large grid-connected PV system as experimental platform, a detailed overview of the PV power plant and environment was firstly given. As further explained, only historic weather related data that could be easily obtained from standard on-site weather sensors was selected to train the models.

Another important aspect of this chapter was the demonstration of a systematic and principled data-science approach towards raw data processing at scale. Finally, with additional feature engineering applied, the data set was ready to be used for model training.

Chapter 3: Forecast model development

As a pre-cursor for the development of the forecasting models, this chapter was used to deliver a brief theoretical background on the basics of ML. This was further continued with the theory related to the FFNN, LSTM-RNN and GRU-RNN models. After establishing the fundamentals, the detailed approach towards model development was presented.

As mentioned in this chapter, forecast practitioners often rely on intuition and an 'organic' exploration of hyperparameters to develop models. This approach is not reproducible at scale and is one of the main identified reasons for the issue of biased and unconvincing claims of model superiority in literature. To address the issue, a heuristic framework of systematic hyperparameter optimisation and forecast model development was proposed. This framework serves to guide future forecasting practitioners towards delivering unbiased forecasting solutions. It further ensures the large scale reproducibility of developing multiple forecast models.

Another unique research contribution was the development of the inverter-clustering technique. Given the extensive computational effort of individually optimising the hyperparameters and parameters of each inverter-level forecast model, a scalable solution was required to reduce this computational expense. After an exhaustive number of iterations applied to various strategies, the inverter-clustering technique was developed. This technique was shown (in Section 4.4.3) to significantly reduce the computational expense of training multiple low-level forecast models. Therefore, this technique serves as a solution, which further enables the scalability and reproducibility of the proposed low-level forecasting methodology for multi-megawatt PV system applications.

Chapter 4: Results

In response to the main research questions, multi-step 1 - 6 h ahead forecasts were delivered for the grid-connected 75 MW rated PV power plant. Given the novel investigation of macro-level vs. aggregated inverter-level forecasts as the primary research objective, the results of both these approaches were directly compared for the selected DL models. By applying the proposed model development framework for hyperparameter optimisation, together with the inverter-clustering technique, two sets of forecasting models were successfully trained, categorised as: 1) macro-level and 2) aggregated inverter-level models.

Macro-level forecasting

As a baseline for comparison, a traditional macro-level approach towards forecasting was applied, with forecasts delivered using stand-alone implementations of state-of-the-art multi-layer FFNN, LSTM-RNN and GRU-RNN models. After an extensive process of training and fine-tuning these models, the GRU-RNN proved to be the most accurate overall macro-level forecast model, with the multi-layer FFNN delivering the second most accurate forecasts. The final results obtained are summarised in Table 5.1 below.

Table 5.1: PV-system macro-level forecast results summary as determined for the test set.

	Models		
Metric	**FFNN**	**LSTM-RNN**	**GRU-RNN**
NRMSE [%]	8.19	8.23	8.12
MAPE [%]	3.53	3.61	3.42

Aggregated low-level forecasting

With the conventional macro-level forecasting accuracies established, the novel analysis was performed of potentially obtaining an enhanced forecasting accuracy from an aggregate of multiple low-level forecasts. This was accomplished by developing individual forecast models for each of the 84 inverters, by applying the model development framework and inverter-clustering technique. The efficacy of the inverter-clustering technique as a scalable and reproducible solution was clearly demonstrated, with the computational expense for inverter-model development reduced by a factor of eight (8x), without sacrificing accuracy.

The final comparison of results obtained with the two forecasting methodologies (macro-level vs. low-level) are summarised in Table 5.2. From these results, the inverter-level GRU-RNN clearly delivered the best overall forecasting accuracy.

Table 5.2: PV-system macro-level vs. inverter-level test-set forecast results summary.

Model	Forecast	NRMSE	MAPE
FFNN	PV-system	8.19	3.53
	Inverter-cluster	8.15	3.45
GRU-RNN	PV-system	8.12	3.42
	Inverter-cluster	8.02	3.39

5.2 Conclusion

Initially a hypothesis was defined, which questioned the ability of a traditional macro-level forecasting methodology to effectively emulate the non-uniform, low-level power output dynamics of large multi-megawatt PV systems. Therefore, it was anticipated that an enhanced forecasting accuracy could be obtained from an aggregate of multiple low-level forecasts, which individually capture these low-level power output dynamics. Given this original research hypothesis, the response to the main research questions are:

Response to research question 1:

> *Is it possible to obtain an enhanced forecast model sensitivity to non-uniform power output dynamics, by using an aggregate of low-level forecasts?*

Yes it is, however, it was underwhelming to see that the aggregated low-level forecasts did not deliver a much more significant performance increase. Nevertheless, at the same time it was impressive to observe the ability of the developed macro-level models to capture and emulate low-level PV-system dynamics at this scale.

Overall, considering the day-time-only results for all weather types, the aggregated inverter-level FFNN model demonstrated the biggest improvement, with the MAPE performance increase over the macro-level forecasts ranging between 0.04 % to 0.4 %, which directly translates to a 30 kW - 300 kW improvement in forecasting accuracy. On the other hand, considering all weather types, the GRU-RNN aggregated inverter-level model delivered a smaller overall MAPE performance increase ranging between 0.03 % to 0.1 %, which translates to an improvement of about 20 kW - 75 kW in accuracy.

Table 5.3: Final day-time-only hourly forecasting MAPE [%] results obtained.

		Time steps					
Model	**Forecast**	**60**	**120**	**180**	**240**	**300**	**360**
FFNN	PV-system	6.03	6.72	7.26	7.60	7.82	8.08
	Inverter-level	5.63	6.50	7.11	7.45	7.79	8.04
GRU-RNN	PV-system	5.77	6.42	6.91	7.37	7.66	7.98
	Inverter-level	5.75	6.33	6.85	7.26	7.58	8.02

Response to research question 2:

With state-of-the-art deep learning models trained as a macro-level forecasting solution, what is the ability of these more powerful models to capture low-level, non-uniform PV-system power output dynamics?

The largest overall performance increase with the aggregated low-level forecasting methodology, relative to the macro-level implementation, was observed for the FFNN model. However, given the small and consistent differences in forecasting accuracy between the inverter-level and macro-level GRU-RNN models, the GRU-RNN macro-level model displayed the best ability to capture the non-uniform, low-level power output behaviour. Overall, the GRU-RNN macro-level model was further shown to also outperform the aggregated inverter-level FFNN model. However, compared to the proposed low-level forecasting methodology, it remains to be proven if the macro-level application of DL-based models can maintain these levels of forecasting accuracy for PV systems larger than the 75 MW system analysed, or utility-scale PV systems with a very uneven surface topology.

Although the inverter-level forecasts delivered marginal improvements compared to the macro-level forecasting approach, no noteworthy improvements were observed regarding the quantified 95 % Bootstrap CI uncertainty distributions. This indicates that the macro-level models have a relatively equal uncertainty in terms of delivering forecasts, as compared to the inverter-level models.

Another important observation was made regarding the holistic response of the overall PV-system to a specific set of input features. It was observed that environmental/atmospheric variables highly correlated with overall PV-system power output are also highly correlated with inverter-level power output. Initially it was hypothesised that the smaller subsets of inverter PV modules would show a different preference for input features. However, it has been shown that the global PV-system power output is indeed a true indicator of the variables, which affect both global-and micro-level power production. The only difference being that the subsets of PV modules (represented by inverter output) are uniquely impacted in terms of the sensitivity to these variables. This observation is of great value to practitioners who are looking to reproduce the suggested inverter-level forecasting methodology for their own applications. The reason being that once the macro-level models have been optimised, the same input feature set and sliding-window size can be used for the inverter-level forecasts, which further reduces the required computational effort of additionally identifying appropriate features for the low-level forecast models.

Finally, the efficacy of DL-based models to emulate the low-level PV-system dynamics has been shown for when the models are applied as a macro-level solution. Therefore, with reference to the broader research community, researchers who have and continue to propose

DL-based forecasting solutions for smaller multi-megawatt (< 10 MW) PV systems, can be confident in the application of their models as macro-level solutions.

5.3 Research contributions

With the research completed, the following contributions have been made:

- A novel investigation has been presented, comparing the traditional macro-level forecasting approach to an aggregated low-level forecasting methodology. The final outcome of this investigation is of great importance to the larger PV forecasting research community. The reason being that it has successfully been demonstrated that macro-level forecasting models, applied by forecasting practitioners, can adequately capture low-level PV system power output dynamics at multi-megawatt scale.
- A unique inverter-clustering technique was further developed, which ensures the reproducibility and scalability of the proposed forecasting methodology. This technique serves to assist other forecasting practitioners who also wish to implement this methodology at multi-megawatt scale.
- To address the discrepancies in literature concerning biased model development, a heuristic process of systematic hyperparameter optimisation was proposed. This framework serves to guide future forecasting practitioners towards delivering unbiased forecasting solutions. It further ensures the large scale reproducibility of developing multiple forecast models.
- Given the lack of published research applied to multi-megawatt PV systems, this research makes an important contribution towards reducing the disconnect between literature and utility-scale PV power production. Specifically, these contributions are related to utility-scale data processing and the deployment of DL models to deliver forecasts at multi-megawatt scale.

5.4 Recommendations

Finally, given the successful execution of the research objectives, the author's advice to PV-forecasting practitioners and PV-system operators is as follows:

1. The proposed low-level forecasting methodology is recommended for PV systems, which produce power at a multi-megawatt (> 10 MW) scale. The reason being that these PV systems have a large ground coverage, which further amplifies the non-uniform exposure of the PV modules, as was for example demonstrated regarding the effects of wind.
2. It is further advised that the aim of an enhanced forecasting accuracy should firstly be pursued with a focus on feature engineering and the appropriate selection of available input features and hyperparameters. Following a systematic process of iterating over various features and hyperparameters, forecasting practitioners are largely guaranteed of obtaining good solutions, as demonstrated with this research.
3. Given the current selection of state-of-the-art DL models and the exhaustive evaluation of these models, the GRU-RNN model is recommended for PV systems in similar

locations regarding environment and weather conditions. However, the use of a multi-layer FFNN remains to be a useful option for forecasting practitioners with limited access to powerful computational resources.

4. As previously mentioned, forecasting practitioners who have and continue to apply their DL models with a macro-level methodology, can be confident in the application of their macro-level forecast solutions for smaller multi-megawatt (< 10 MW) PV systems.

5.5 Future research

With the research completed, other research topics, remaining evaluations, and open questions have been identified and are outlined as:

- Apart from DL, reinforcement learning is an emerging research hotspot within the field of PV-power forecasting. By dividing the PV system into smaller segments, different agents can be allocated to each segment and trained to capture the underlying dynamics. Given the success of reinforcement learning, this is a very promising and relatively unexplored research area within PV forecasting as a research field.
- With solar power becoming more accessible to the public, small grid-connected rooftop PV systems are growing in popularity. Given this effect of multiple smaller PV systems, located in close proximity within a city or large neighbourhood area for example, aggregated low-level forecasting has great potential to further enhance the predicted PV power output (and therefore load demand). For example, should each low-level model represent a single rooftop PV system, with a unique orientation and exposure to the environment, then models trained for each of these small subsets of modules are anticipated to deliver a much more accurate prediction of power output than a conventional macro-level forecasting approach.
- With the availability of large open spaces considered to be a scarcity in some countries, an aggregated low-level forecasting analysis could prove to be of great value to PV systems constructed on complex terrains. An example of this is demonstrated with Figure 5.1, which is representative of the Taihang Mountain PV system in Handan China [139]. PV systems located in such environments with hills and uneven ground distributions are expected to have exaggerated non-uniform power output distributions, caused for example by wind movement and received irradiance at different angles of incidence. Research applied to these complex terrains would be very interesting.

Figure 5.1: Example of a large PV system constructed on complex terrain

Appendices

Appendix A

PV-system data

A.1 Available weather sensor data

The list of available weather sensor data, as captured at various locations within the PV system, are presented in Table A.1

Table A.1: Weather sensor data recorded.

Weather sensor data			
Measurement	Unit	Measurement	Unit
Absolute-Air-Pressure	[hPa]	Global-Diffused-Irradiance	[W/m^2]
Rain-Intensity	[mm/h]	Global-Horizontal-Irradiance	[W/m^2]
Relative-Humidity	[%]	Pyranometer-Irradiation-InCline	[W/m^2]
mean wind direction	[0° − 360°]	Ref-Cell-Irradiation-InCline-1	[W/m^2]
mean wind speed	[m/s]	Ref-Cell-Irradiation-InCline-2	[W/m^2]
Pyranometer-Irradiation-Horizontal	[W/m^2]	Environment-temperature	[C°]
Avg-Direct-Normal-Irradiance	[W/m^2]	Module-temperature	[C°]
Global-Diffused-Irradiance	[W/m^2]	Direct-Normal-Irradiance	[W/m^2]
Avg-Global-Horizontal-Irradiance	[W/m^2]	Pyranometer-Calculated-Incline	[W/m^2]

A.2 PV-system power output related data

The data points measured by each inverter are summarised in Table A.2.

Table A.2: Inverter sensor data recorded.

Inverter sensor data			
Measurement	Unit	Measurement	Unit
DC current input	[A]	AC switch inverter	
DC input voltage	[V]	Grid switch inverter	
DC power input	[W]	Time until grid connection	[s]
AC current output line 3	[A]	Error code inverter	
AC current output line 2	[A]	Recommended actions	
AC current output line 1	[A]	Insulation resistance	[Ohm]
AC current total output	[A]	External temperature (air supply)	[C°]
AC voltage output phase CA	[V]	Internal temperature 1	[C°]
AC voltage output phase BC	[V]	Operating hours internal fan 1	[s]
AC voltage output phase AB	[V]	Internal temperature 2	[C°]
Total AC active power output	[W]	Operating hours internal fan 2	[s]
Total AC energy delivered per day	[Wh]	Heat sink temperature	[C°]
Total AC energy delivered	[Wh]	Operating hours heat sink fan	[s]
Apperant AC power	[W]	Temperature2 sine-wave filter	[C°]
Reactive AC power	[VAR]	Operating hours internal heater	[s]
Frequency of delivered power	[Hz]	Event ID of the current event	
Active power target value	[W]	Serial number string monitor controller	
Active power target	[%]	Operation state string monitor controller	
Reactive power target value	[Var]	Operation hours string monitor controller	[s]
Reactive power target	[%]	Warning code string monitor controller	
Displacement power factor active/reactive		String monitor N current channel 1	[A]
Operation mode active power		String monitor N current channel 2	[A]
Operation mode reactive power		String monitor N current channel 3	[A]
Operating state status		String monitor N current channel 4	[A]
Operating state mode		String monitor N current channel 5	[A]
Operating hours	[s]	String monitor N current channel 6	[A]
Feed-in hours	[s]	String monitor N current channel 7	[A]
DC switch inverter		String monitor N current channel 8	[A]

The same data points are used as measured by the LV/MV and MV/HV transformers, which are displayed in Table A.3.

Table A.3: Available data points recorded for HV and MV transformers.

HV & MV transformer sensor data			
Measurement	Unit	Measurement	Unit
Active Energy Consumption	[Wh]	Reactive Energy Consumption	[VARh]
Active Energy Feed In	[Wh]	Reactive Energy Feed In	[VARh]
Active Power Consumption	[W]	Reactive Power Consumption	[VAR]
Active Power Feed In	[W]	Reactive Power Feed In	[VAR]

The power transducer captures important AC related power output values. These captured data points are all summarised in Table A.4.

Table A.4: Sensor data captured by power transducer.

Transducer sensor data			
Measurement	Unit	Measurement	Unit
Current line A	[A]	Total harmonic distortion of voltage phase C	[%]
Current line B	[A]	Total harmonic distortion of Current phase A	[%]
Current line C	[A]	Total harmonic distortion of Current phase B	[%]
Current Average	[A]	Total harmonic distortion of Current phase C	[%]
Current Average min	[A]	Current K-Factor phase A	
Current Average max	[A]	Current K-Factor phase B	
Current Average mean	[A]	Current K-Factor phase C	
Frequency	[Hz]	Current Crest Factor phase A	
Frequency min	[Hz]	Current Crest Factor phase B	
Frequency max	[Hz]	Current Crest Factor phase C	
Frequency mean	[Hz]	Voltage Flicker N phase A	[V]
Current unbalanced	[A]	Voltage Flicker N1 phase A	[V]
Voltage line A	[V]	Voltage Flicker N phase B	[V]
Voltage line B	[V]	Voltage Flicker N1 phase B	[V]
Voltage line C	[V]	Voltage Flicker N phase C	[V]
Voltage line Average	[V]	Voltage Flicker N1 phase C	[V]
Voltage line Average max	[V]	Voltage Harmonic N phase A	[%]
Voltage line to line phase AB	[V]	Voltage Harmonic N1 phase A	[%]
Voltage line to line phase BC	[V]	Voltage Harmonic N2 phase A	[%]
Voltage line to line phase CA	[V]	Voltage Harmonic N phase B	[%]
Voltage line to line Average	[V]	Voltage Harmonic N1 phase B	[%]
Voltage line to line Average max	[V]	Voltage Harmonic N2 phase B	[%]
Voltage line to line Average mean	[V]	Voltage Harmonic N phase C	[%]
Total Active power	[kW]	Voltage Harmonic N1 phase C	[%]
Total reactive power	[kVAR]	Voltage Harmonic N2 phase C	[%]
Active energy delivered	[kWh]	Voltage Inter-harmonic N phase A	[%]
Active energy received	[kWh]	Voltage Inter-harmonic N1 phase A	[%]
Reactive energy delivered	[kVARh]	Voltage Inter-harmonic N phase B	[%]
Reactive energy received	[kVARh]	Voltage Inter-harmonic N1 phase B	[%]
Power factor		Voltage Inter-harmonic N phase C	[%]
Total harmonic distortion of voltage phase A	[%]	Voltage Inter-harmonic N1 phase C	[%]
Total harmonic distortion of voltage phase B	[%]		

A.3 Data acquisition and storage

With the data received, the raw CSV file data is structured. This is done by uploading the data to a database, which allows queries to be executed and further data processing to be performed effectively. The database is established on a QNAP manufactured Network Attached Storage (NAS) drive, illustrated in Figure A.1, and is managed by means of the MariaDB relational database management system (RDBMS). MariaDB (similar to MySQL as a RDBMS) allows the data to be accessed from database queries executed within a python script, or by means of an open source administration tool such as phpMyAdmin. During the execution of this research, database information is primarily obtained from queries executed in python scripts.

To avoid the loss of data, due to a hard-drive malfunction, two 6 TB hard drives are configured with a Raid-1 configuration, which essentially allows the main hard-drive to be mirrored

(replicated) onto the second hard drive. The MariaDB database structure, as hosted on the NAS drive is configured as displayed in Figure A.2

Figure A.1: QNAP NAS drive.

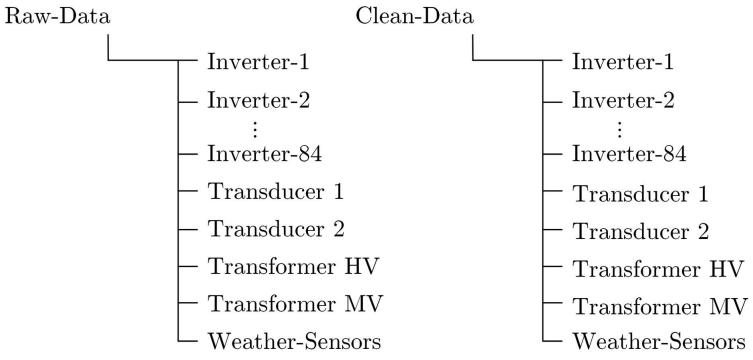


Figure A.2: Raw and clean data database structure.

To further improve the speed of database queries, the cleaned database is deployed directly onto the server. This is done by duplicating the database as a MariaDB Docker container. The reason for this setup is due to the deployment of ML models on a remote server, running within Docker containers. Docker is specifically used to ensure a stable development environment, where all of the dependencies required to execute the ML model code are maintained. With container linking, the Docker container within which the ML code is executed, is linked to the Docker container, which houses the database. This is illustrated in Figure A.3. The use of Docker containers provided a significant project advantage regarding code reproducibility, consistency, maintainability and remote deployment.

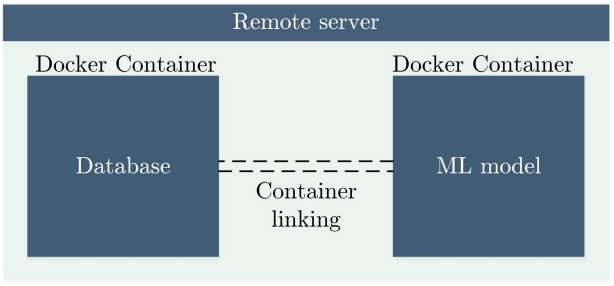


Figure A.3: Deployment of database onto docker container and linking with ML models hosted on another Docker container.

A.4 Data processing

A.4.1 Data resolution

Before this section continues with the data pre-processing steps applied, it should be mentioned that the acquired data set consists of 49 months of entries, all recorded at 1 min resolution. However, this high resolution is found to be redundant, with a 5 min data resolution more than capable to capture the dynamics of the environment and the intrinsic behaviour of the PV system in response to the external conditions. Evidence regarding the adequacy of a 5 min resolution is displayed in Figure A.4, which illustrates an irradiance time-series data signal for days varying in terms of dynamic behaviour.

The reduced data set resolution further aids with GPU memory requirements for the training process of the forecasting models. With four years of available data, this reduced 5 min resolution speeds up model training in general, while still maintaining enough data for efficient model training.

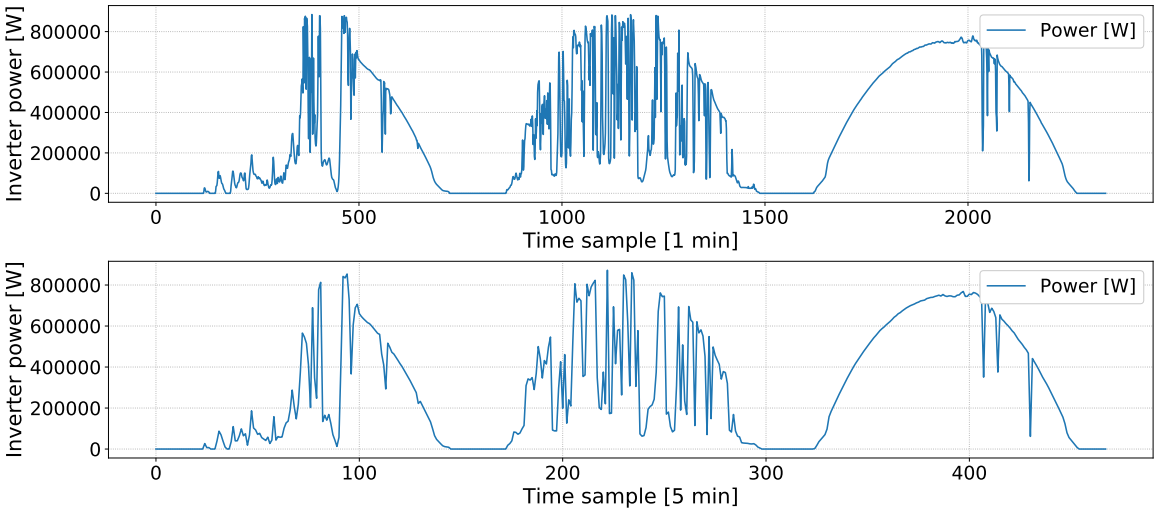


Figure A.4: Data set resolution comparison. Top: Data presented at 1 min resolution. Bottom: data presented at 5 min resolution.

As described in Section 2.5.1, the first step towards obtaining a representative data set is to eliminate all data entries outside of the predefined boundary conditions. This is followed by the task of interpolating small segments of missing data entries and imputing large segments of missing data.

A.4.2 Data pre-processing: Interpolation

The execution of the interpolation strategy is done based on three points of criteria.

Firstly, all data points that represent a component of power and irradiance are set equal to zero for hours when there is an absence of sunlight (night time, early morning, late afternoon). These hours are defined as 20:00 to 05:00, which is determined from the historic data set entries where the latest and earliest sunshine was available. The remaining weather data entries are excluded from this interpolation, since night-time weather measurements, such as temperature and wind speed might adopt other values than zero.

Secondly, all missing data is interpolated with a linear function. As presented with Figure A.5, the use of polynomial interpolation results in an unrealistic data representation, as inferred from previous data values. For example, the irradiance data in Figure A.5 is unrealistically interpolated below zero. Polynomial interpolation is found to be adequate for clear days, which have little to no high frequency dynamics. However, data values which have high intermittence are found to be misrepresented by the polynomial interpolation, which is why linear interpolation is found to be the best approach.

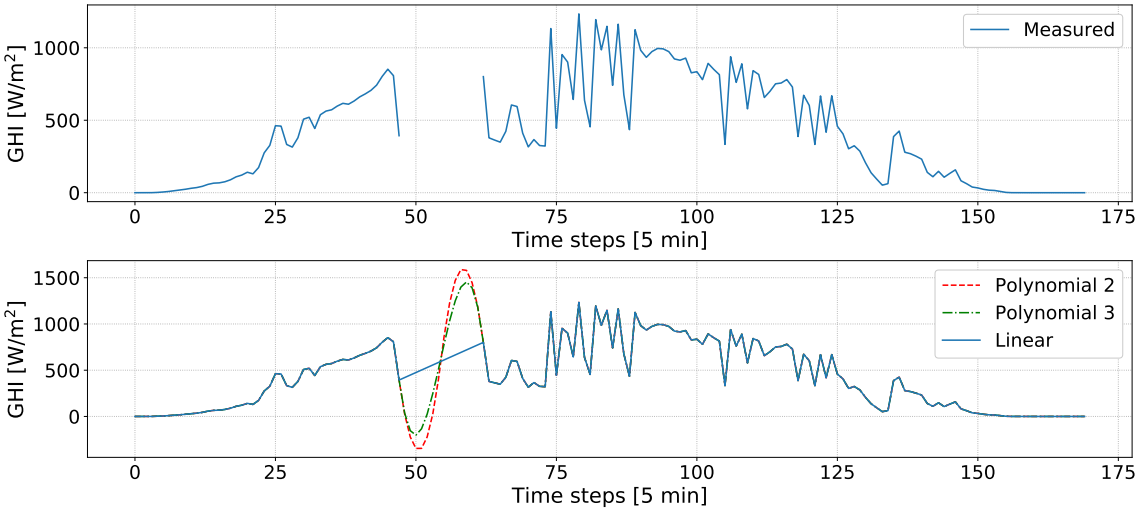


Figure A.5: Example of linear and polynomial interpolation applied.

With linear interpolation selected as strategy to impute missing data points, the maximum number of data points must still be defined, which are allowed to be replaced with real values. Should interpolation be applied in a scenario with too many missing data points, a bad representation of true data will be obtained, which might impede forecasting model accuracy. As a reference to define the boundary condition within which interpolation is allowed to be applied, the dynamics of a clear-sky day is evaluated. The reason being that, the dynamics of the power output data points (ultimately to be predicted by the forecasting model) is regarded as the baseline trend of power output for any day type. Also, on a clear day, the power output is most predictable due to the known movement of the sun. Figure A.6 illustrates the process of defining a maximum missing data time period for the condition of linear interpolation.

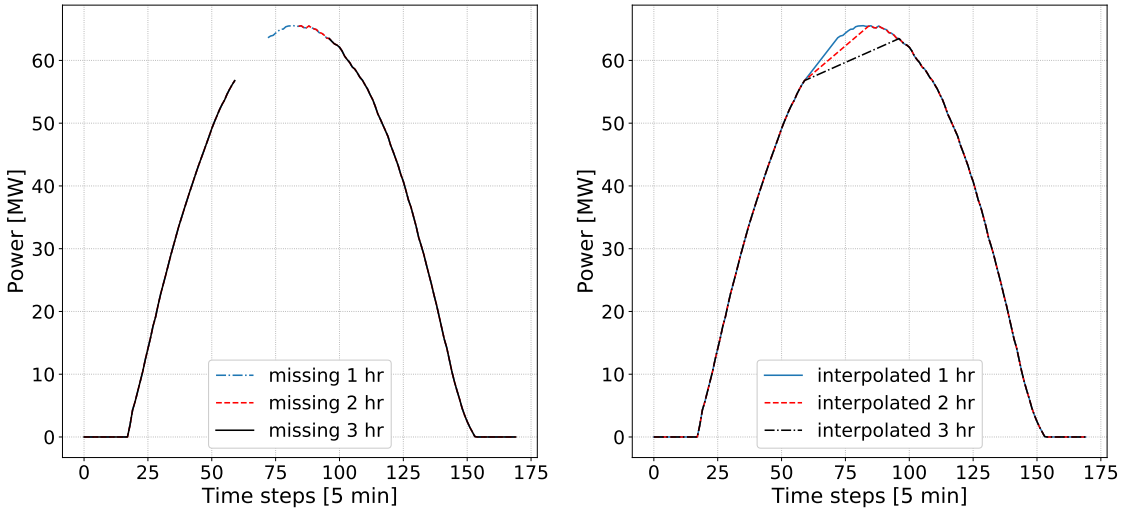


Figure A.6: Example of linear interpolation applied to data missing for a 1 h, 2 h and 3 h period.

As an example for the given graph in Figure A.6, the maximum error made due to the interpolation is illustrated in TableA.5. Of course, the error will be much less, for the early and late hours of the day, since power output is almost a straight line for these segments. However, it is during the peak of the day where the highest power output is delivered, which also leaves room for the largest error to be made.

Table A.5: Error correlation of number of missing data points and values linearly interpolated.

Hours of missing data [h]	Maximum interpolation error [%]
1	1.01
2	3.74
3	7.57

From this analysis it is decided that, continuous data segments with more than 1 h of missing data are not interpolated. The application of this interpolation strategy is once again presented with Figure A.7, in the the bottom graph. From this graph it is evident that three of the four days, which contained missing data points, are kept within the data set due to successful interpolation. Only one day, which occurs between the two dashed vertical lines could not be successfully interpolated and is therefore eliminated from the data set.

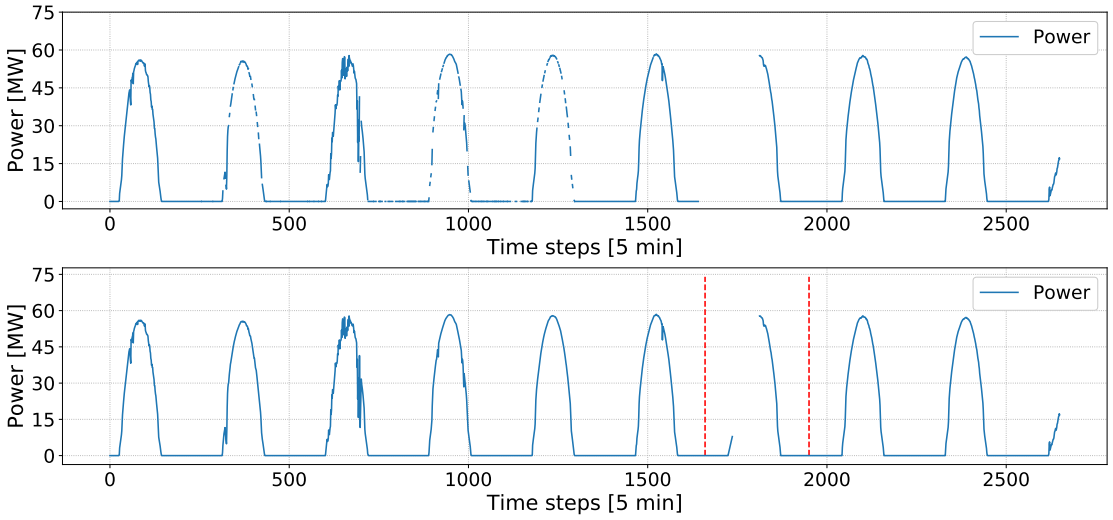


Figure A.7: Example of linear interpolation strategy applied to missing data segments.

To be clear, if any of the sensor data entries, utilised for model training, have more than 1 h of continuously missing data, all data set entries associated with this particular day are rendered useless and are disregarded. This is demonstrated in Figure A.8, where too many continuous data entries are missing for the temperature sensor, which therefore results in the GHI and wind data also being dropped from the final data set used for model training. From this, the necessity of a continuous data set should be evident.

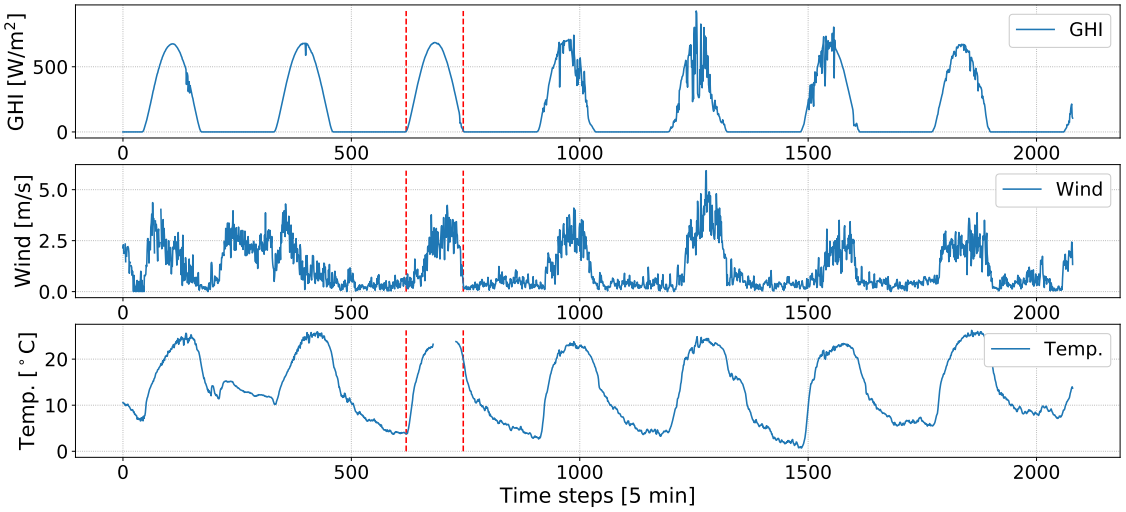


Figure A.8: Example of data elimination where one of the input features has more than 1 h of missing data.

A.4.3 Data pre-processing: Imputation

For any utility-scale PV system, with hundreds of sensors collectively working together to measure data, large (weeks, months) segments of continuous missing data entries are a certainty. As mentioned, the primary objective with data imputation is to substitute these

large segments of missing/faulty data entries with another representative set of data points. In particular, data imputation is executed for the weather and PV-system power data sets.

A.4.3.1 Weather data

Initially, it is considered to train models with the weather station data, together with the power output of the inverters closest to the various weather stations. Reason being to obtain a more localised weather data representation within the large PV system. However, in some cases weather stations are too far away from inverter blocks, to confidently associate the two data sources with one another. It is also seen within the data set that several of the weather sensors have missing data segments, ranging between a few weeks to a few months. This further disqualifies the suggested approach of pairing specific weather sensor and inverter power data to train the forecast models.

Therefore, a decision is made to instead use the average value for the recorded weather data variables. An average measurement for each weather-based variable is determined from the aggregate of all five weather sensors. This data is then used as input for the training of the models. However, before the weather data set is established, it is necessary to process and validate the data.

Weather data: Temperature data

Temperature data will be used as input feature for model training. However, before the temperature data set is established from the average of all the weather sensors, the data signals are first to be inspected to avoid including data deficiencies. This process of eliminating bad sensor measurements is illustrated in Figures A.9 - A.11. With Figure A.9, all of the available temperate data sets from the various weather sensors are displayed, except for the WS-CB sensor, which did not retrieve any temperature data.

From Figure A.9 the temperature deviation of weather station WS-TR15, is not completely obvious. Therefore, to assist with identifying any sensor deviations, an average is determined from the four different weather stations. The difference between the averaged temperatures and the individually recorded temperatures of each sensor is then determined. These differences are illustrated in Figure A.10, from which the data of WS-TR15 is evidently not only missing for a few months, but also deviates somewhat compared to the other temperatures, as indicated for the section between the vertical dashed lines. To avoid any uncertainty, this segment of data is eliminated.

With no further temperature data deviations observed, a new average (which ignores missing data points) is determined as presented with Figure A.11. This final averaged temperature data set (WS-Avg.) is used for training the forecast models.

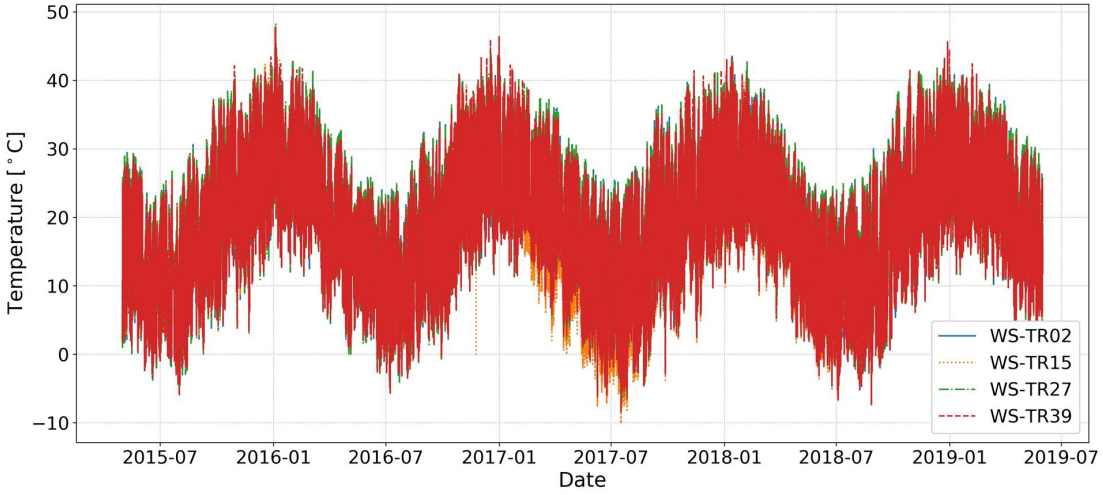


Figure A.9: Measured temperature data from multiple weather sensors.

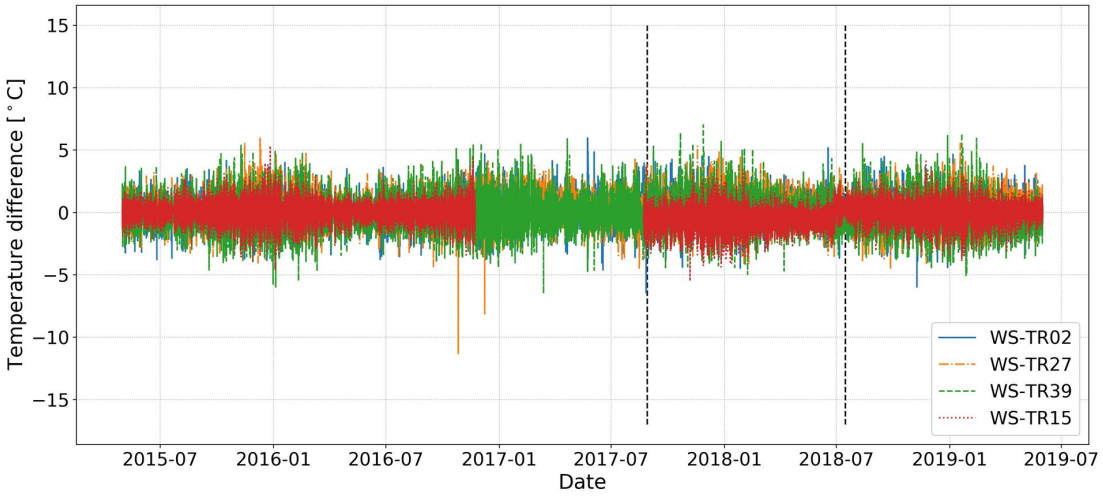


Figure A.10: Illustration of missing and a prolonged temperature data deviation from group average for weather sensor WS-TR15.

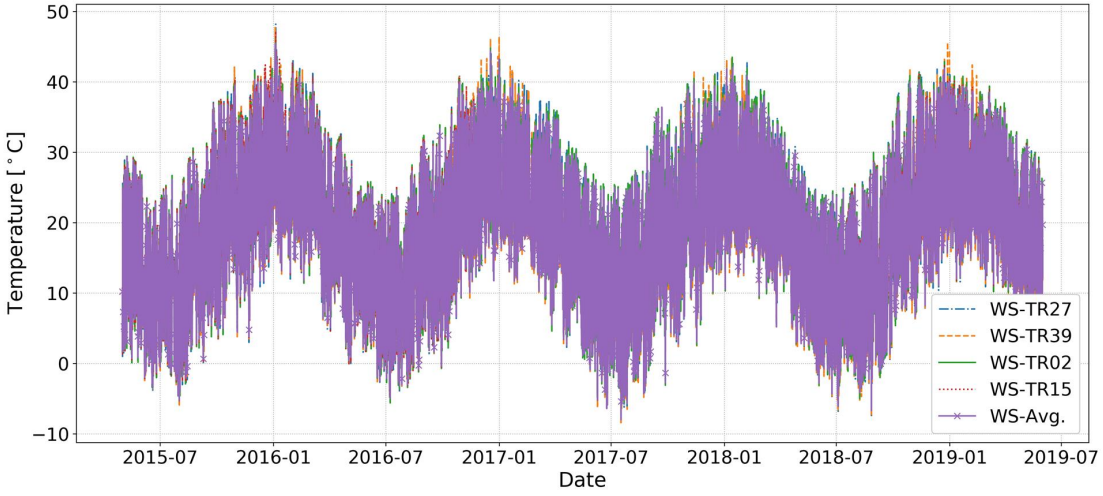


Figure A.11: Final averaged temperature sensor data set after all data deviations are eliminated.

Weather data: GHI data

The recorded irradiance data from all five weather stations is presented with Figure A.12. From closer inspection, it is determined that the irradiance data measured at the control building (WS-CB) consistently deviates from the group norm, as illustrated in Figure A.13 for the period between 2017-11 to 2018-02. This deviation is found to occur for other large segments of the entire data set (can be observed in Figure A.12), which is why GHI data from weather station WS-CB is eliminated as a whole. After removing the deviating GHI data, a new average is determined for the remaining four weather stations, from which the differences are once again determined and plotted as presented with Figure A.14. The large spikes observed from this graph are temporary and typically occur during intermittent days, where clouds partially shade the PV system and the distributed weather stations. With no other data deviations observed, Figure A.15 demonstrates the final GHI data set (WS-Avg.) used for model training.

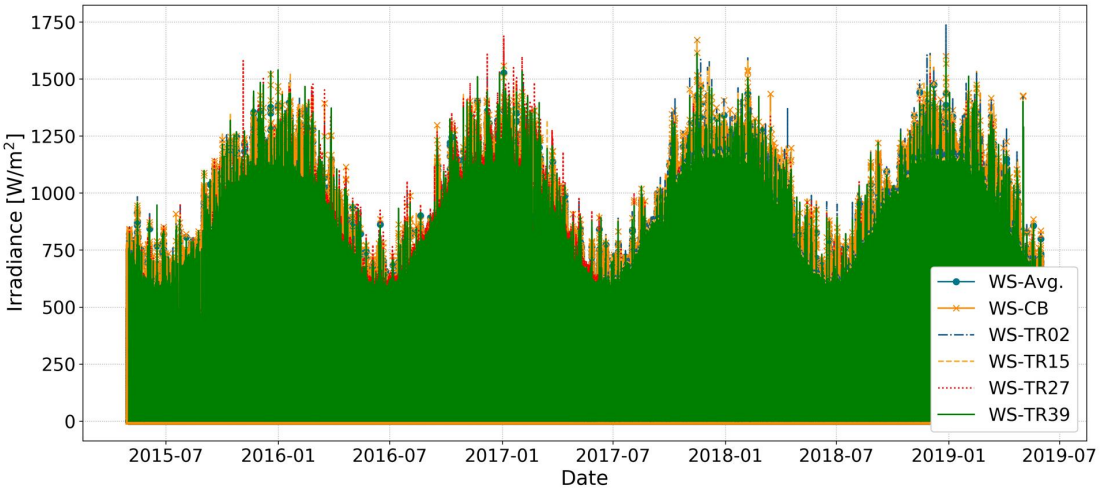


Figure A.12: Measured GHI data from multiple weather sensors.

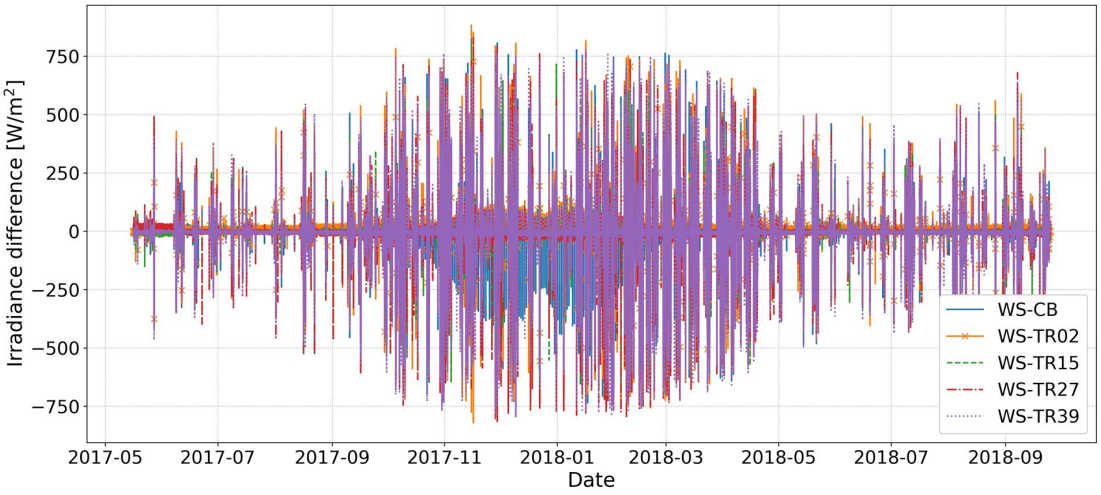


Figure A.13: Illustration of constant irradiance data deviation from group average for weather sensor WS-CB.

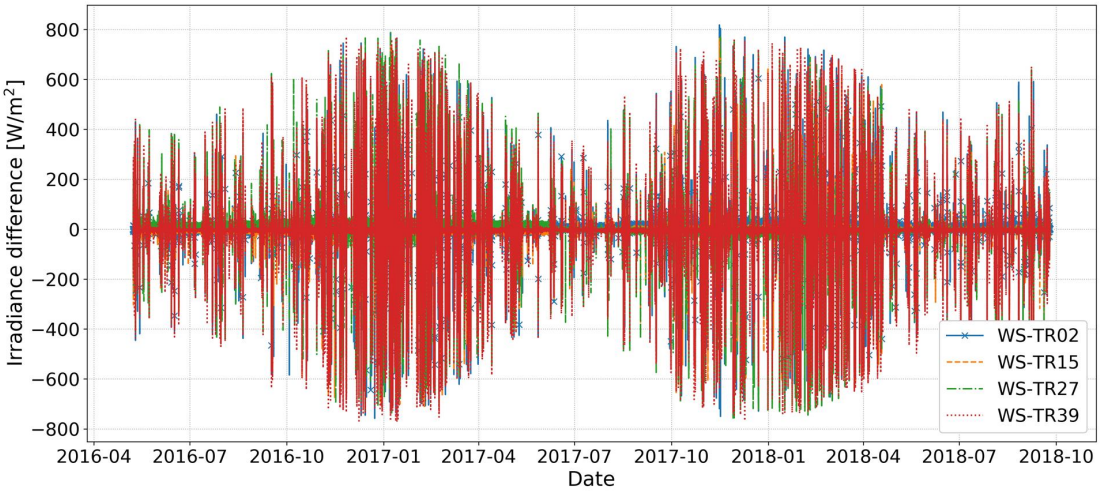


Figure A.14: GHI data set differences relative to group average for all weather sensors after eliminating WS-CB weather sensor data.

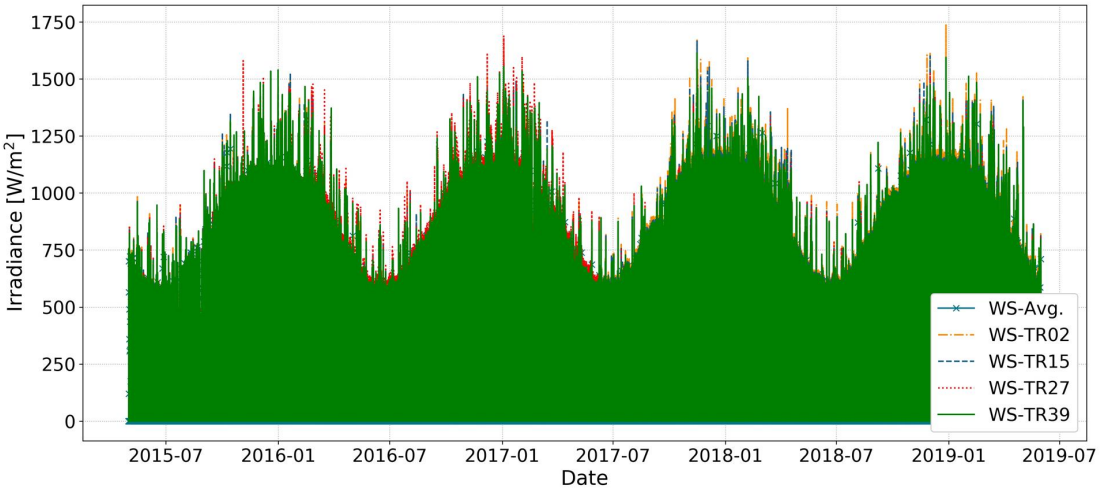


Figure A.15: Final averaged GHI data set used for model training.

Weather data: Wind speed data

Wind-speed measurements are recorded by all five weather stations. It is decided to also include the wind data from the control building weather station (WS-CB). Although at approximately double the height as the other weather stations, this weather station does provide an added perspective regarding overall wind movement slightly above the PV system. The collection of all the wind speed data sets are plotted in Figure A.16.

Immediately evident from Figure A.16 is the large wind speed values of weather station WS-TR15, relative to the group. With further investigation, the deviation of the measurements for WS-TR15 from the group norm is demonstrated in Figure A.17.

From Figure A.17, it is evident that wind speed for sensor WS-TR15 demonstrates abnormalities for the period of 2015-05 to 2016-03, since it is much higher than the norm. Also, there is an irregular zero value movement and a missing segment of data for the period between 2016-3 to 2018-1. Only after 2018-1 does the behaviour of WS-TR15 re-align with the group norm. Therefore, all data for weather sensor WS-TR15 is removed up to 2018-1, as demonstrated with Figure A.18.

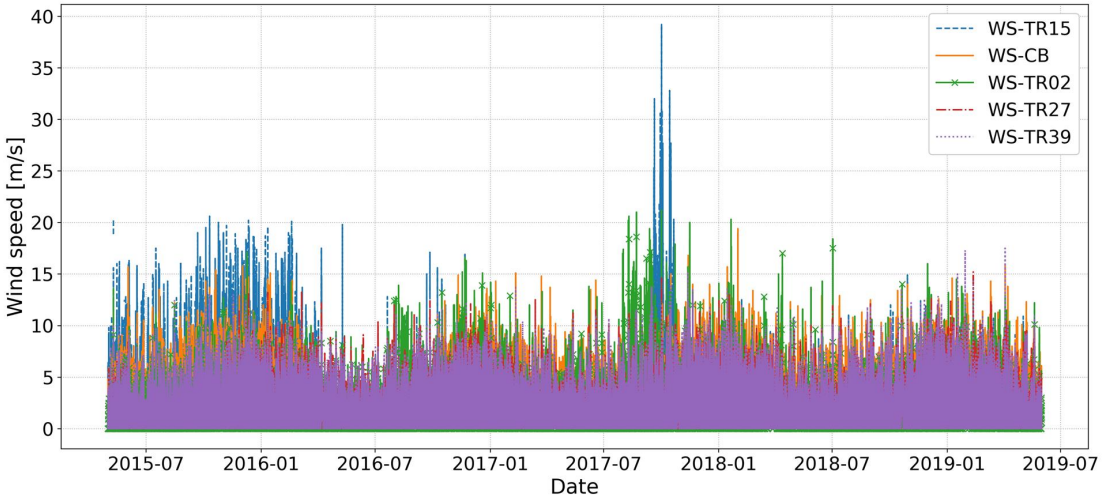


Figure A.16: Wind speed measurements of all weather stations.

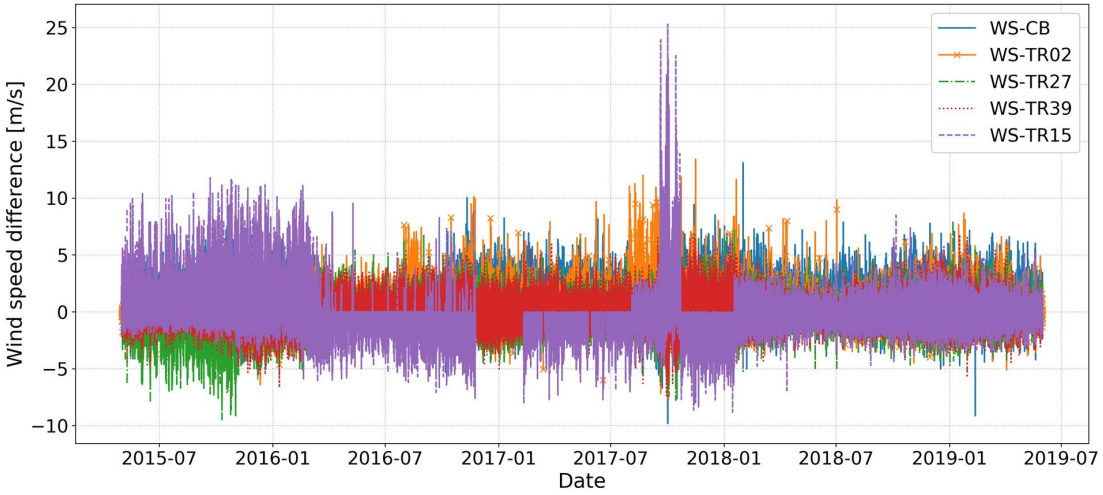


Figure A.17: Wind speed differences relative to group average for all weather stations.

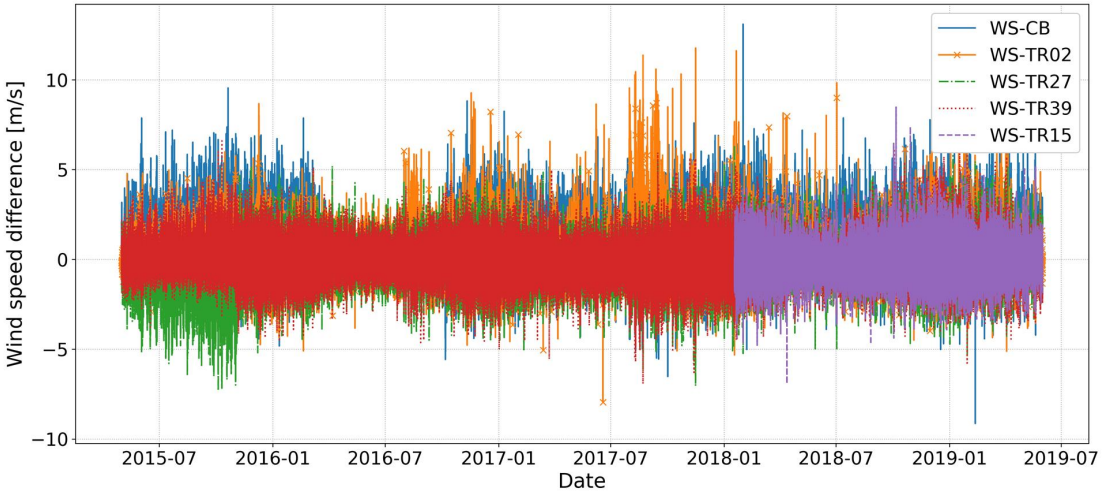


Figure A.18: New wind speed differences relative to group average for all weather stations after the elimination of WS-TR15 for the period of 2015-05 to 2018-01.

With the data segment removed of WS-TR15, a new average is determined, from which further weather sensor deviations are evident from Figure A.18. From this graph, wind speed sensor WS-TR27 has consistently recorded low values, relative to the group norm, for the period of 2015-5 to 2015-11. Also, WS-TR02 has a high difference for the period of 2017-08 to 2017-10.

From further analysis, as presented in Figure A.19, weather station WS-TR27 evidently has an irregular output from 2015-5 up to 2015-11. This segment of data is removed for WS-TR27, from which sensor differences relative to the new group norm are once again determined, as displayed in Figure A.20. From the graph in Figure A.20, it is not evident that the WS-TR02 wind speed for the period between 2017-7 and 2017-10 is to be regarded as irregular. Gust winds are a possible cause of this behaviour. Also, given the position of weather station WS-TR02, which is on the Northern outer part of the PV-system, the increased wind speeds for the period are a further possibility. For these reasons, the consistent difference in wind speed data of WS-TR02, for the period of 2017-7 and 2017-10, is not regarded as irregular enough to be eliminated. Finally, the averaged wind speed data set (WS-Avg.) to be used for model training, is presented in Figure A.21.

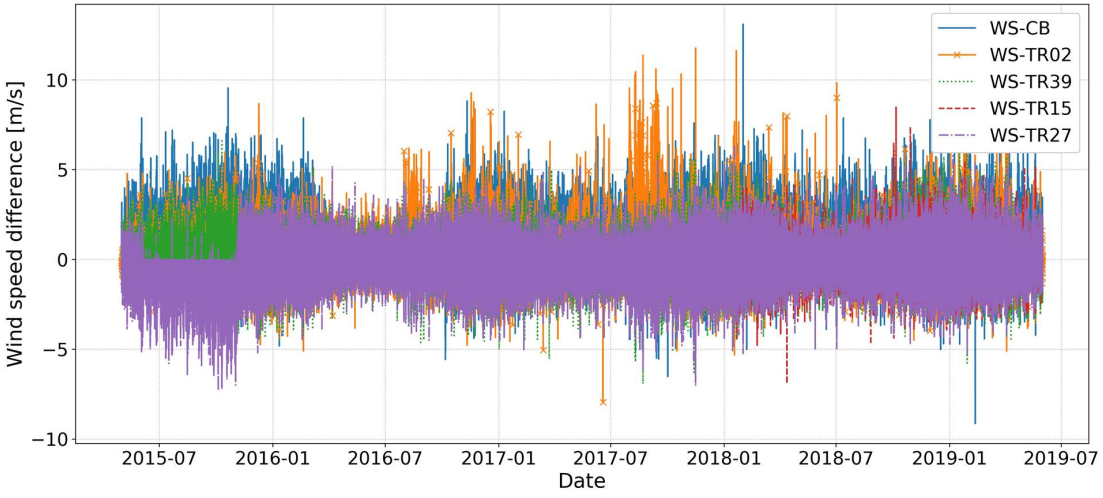


Figure A.19: Observed wind speed measurement deviation of WS-TR27 from group norm.

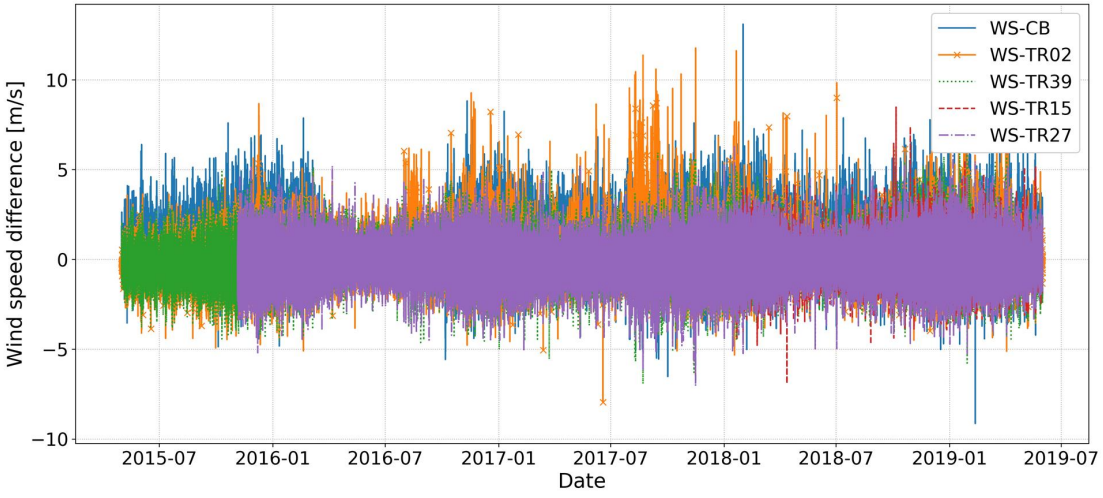


Figure A.20: Wind-speed differences from group norm, after the elimination of WS-TR27 data segment for 2015-5 to 2015-11.

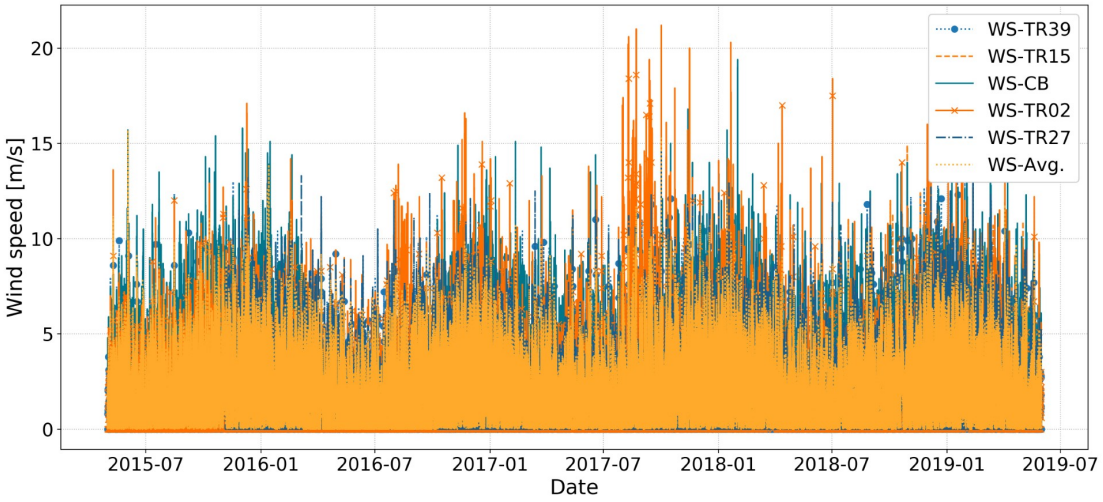


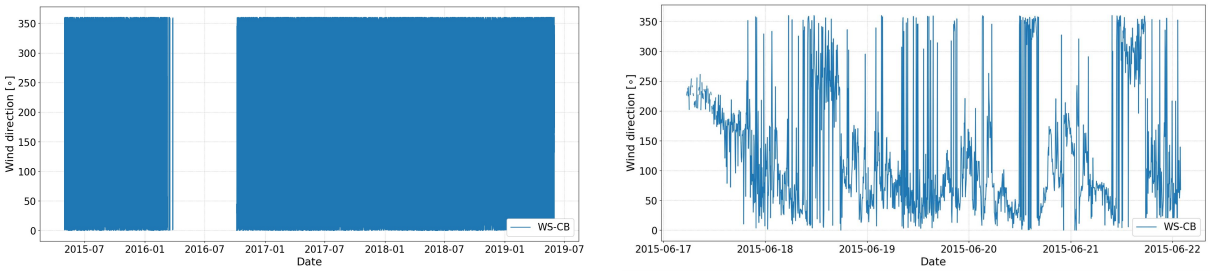
Figure A.21: Final averaged wind speed data set.

Weather data: Wind direction, air pressure, humidity data

Only weather station WS-CB recorded measurements of the WD, AP and RH. The measured WD values, range between $0° - 360°$, with the entire data set presented in Figure A.22. Evident from the data set is that a large segment of WD data is missing for 2016-3-8 to 2016-10-6.

The values recorded for AP, shown in Figure A.23, typically range between 870 - 900 hPa. There is a clear seasonality regarding these measurements, with values high in the winter months and lower for the summer months. The risk of having only one sensor available for a particular measurement is once again evident from the large segment of missing data as seen in Figure A.23 for the period of 2016-3-8 to 2016-10-6.

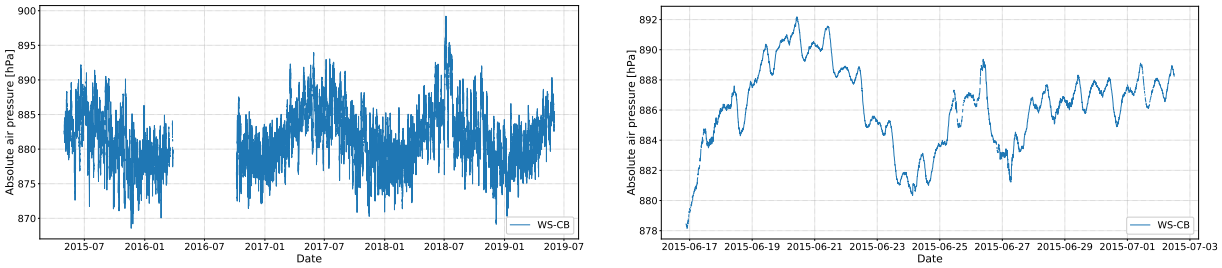
Also demonstrating a clear seasonal trend is the RH measurements, which range between 0 - 100 %, as evident in Figure A.24. With WD, AP and RH values all recorded by a single weather station, all three of these data sets have a large segment of missing data for the period of 2016-03-08 to 2016-10-06. As mentioned, it is required to expose the models trained to complete data sets, which testifies to the generality of the forecast models.



(a) Entire wind direction data set.

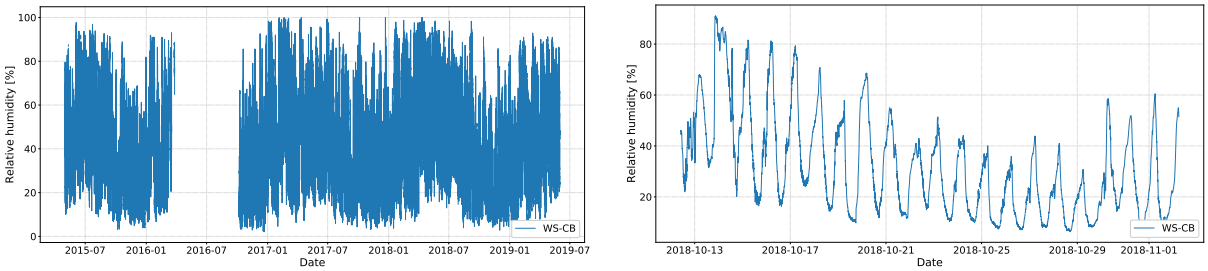
(b) Wind direction detailed.

Figure A.22: Wind-direction measurements.



(a) Entire absolute air pressure data set. (b) Absolute air pressure detailed.

Figure A.23: Absolute-air-pressure measurements.



(a) Entire relative-humidity data set. (b) Relative humidity detailed.

Figure A.24: Relative-humidity measurements.

Fortunately, another research weather station was operational on site for this period, located at close proximity from the control building. This weather station was used for an on-site research project, also lead by Stellenbosch University, which measured WD, AP and RH. With the raw weather data available at 1 min intervals and the calibration dates of the weather station validated, the raw data is processed and outlier data points eliminated.

Importantly, the obtained data is also corrected to eliminate any offset to that of the WS-CB measurements. This data correction is applied with more than 2000 data points using a linear regression model optimised with the python-based Scikit-learn module. With the linear regression model applied to the research weather station data for the missing period of 2016-03-08 to 2016-10-06, values are imputed for the missing data segments, resulting in the more complete data sets illustrated in Figure A.25.

From the replaced data, there is evidently a small dip in the maximum values for the WD data. This is because the research weather station WD data is labelled according to the 16 wind directions (N, NNE, NE, etc.) and a raw angle measurement is not available. Wind-direction values are therefore divided as intervals of 22.5°. However, since only eight wind directions (N, NE, E, etc.) are considered for this research, North is indicated with the values ranging from 337.5° to 0° and 0° to 22.5°. Thus, with WDs now only available at 22.5° resolution, the max value is 337.5°, which explains the dip seen in the WD data. However, this low resolution of 22.5° is not an issue, since these values are one-hot encoded for use by the ML models, as discussed in Section 2.6.1.3.

From the data presented above, it is evident that not all weather stations continuously, without fault, delivered measurements for the entire time period. Fortunately, with interpolation and imputation strategies, full weather data sets could be obtained, so that the forecast models can be exposed to full cycles of seasonality during training and evaluation, which further strengthens the confidence of the delivered results of this dissertation.

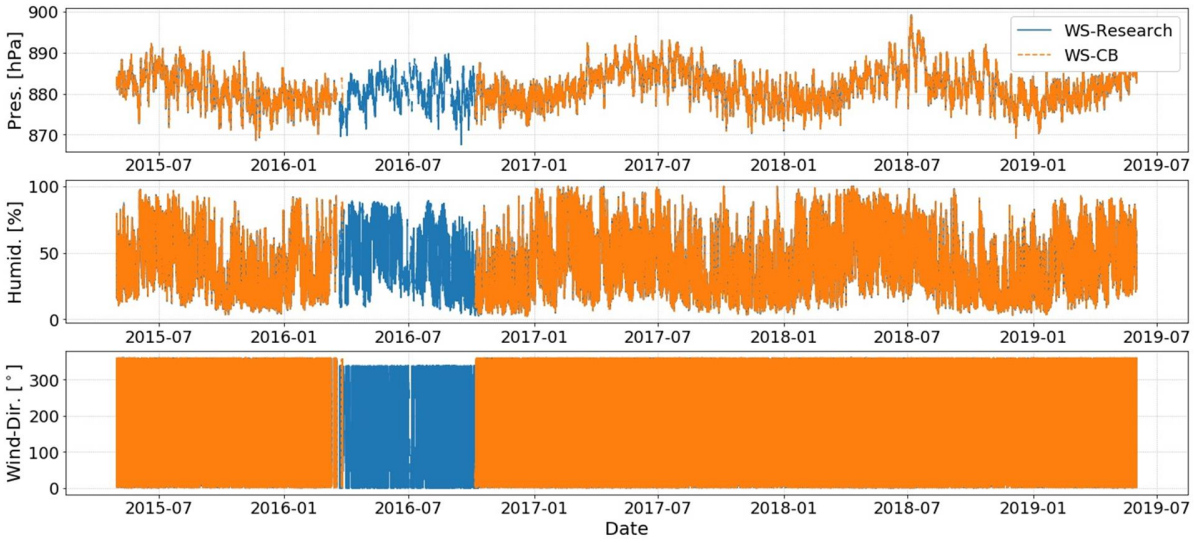


Figure A.25: Missing weather station data imputed.

A.4.3.2 PV-system power data

With the purpose of this research to compare macro-level and aggregated inverter-level forecasts, there are effectively two power output data sets used, as demonstrated with Figure A.26. The inverter-level models are trained with the power data obtained for each inverter, whereas the macro-level PV-system models are trained on the power data, representative of total PV-system power output. Importantly, as mentioned, both processes of macro-level PV-system and inverter-level model development are still executed with the same exogenous input features, such as temperature, irradiance, etc.

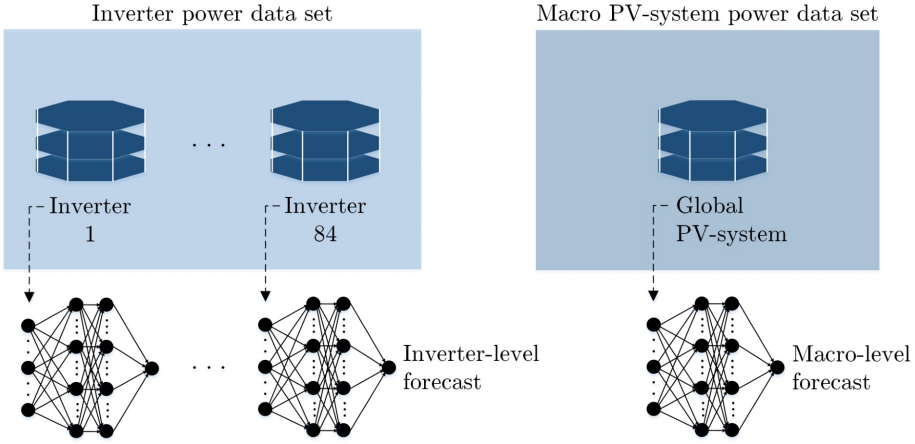


Figure A.26: Distinction between the inverter data sets used for inverter-level models and the total PV-system power output data set used for macro-level forecast model development.

With the inverter power output feeding into the transformers, the global PV-system power output data is directly correlated to the power output of all 84 inverters. This relationship is defined by Eq. (A.1), where P_{macro} represents the macro power output and P_{inv-i} the power output of a single inverter.

$$P_{macro} = \sum_{i=1}^{84} P_{inv-i} \tag{A.1}$$

Therefore, to deliver a fair and accurate comparison of ML model forecasting ability at macro-level and inverter-level, this relationship must be accounted for during the process of preparing the power related data.

After the interpolation of the inverter and PV-system power data sets, there are still bad data segments that need to be addressed, which are categorised as:

- Large segments of missing data.
- Power output failures, logged as 0 W.

Regarding the large segments of missing inverter-power data, the missing data of each inverter and the macro PV system data set is displayed as a percentage value of total data entries in Figure A.27.

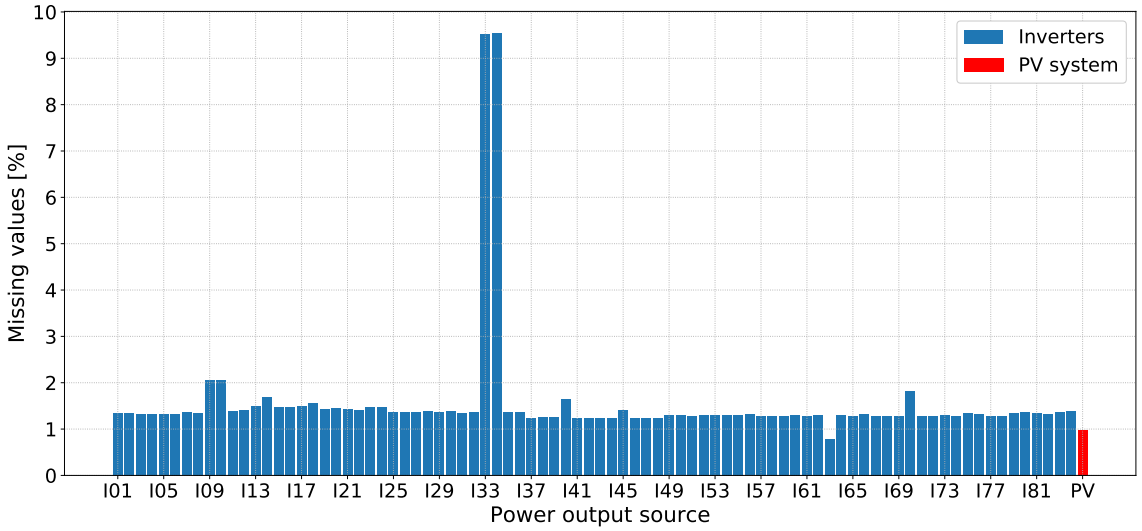


Figure A.27: Percentage of missing data for each inverter and the macro PV-system power data sets.

From the graph of Figure A.27, it is evident that there are only a few inverters with above average (1.3 %) missing data points. Inverters I33 and I34 have the most missing data entries. As mentioned, model development is impaired significantly for scenarios where there are many consecutively missing data points, which leads to under-trained models for specific time periods (certain months or seasons). Since these missing values can either be dispersed across the entire data set (has less impact on model training), or could have occurred consecutively (will impact model development drastically) it is necessary to identify the most consecutive missing data entries for the inverter and macro power data sets.

From analysis, the most consecutive missing data points for both the macro and inverter power data sets are displayed in Figure A.28. A detailed summary is further presented in Table A.6, which shows that inverters I33 and I34 have the longest uninterrupted data set entries of missing values. From investigation, this is attributed to a lightening strike that damaged the inverters, which resulted in more than four months of missing data. The second largest consecutive missing data segments are identified for inverters I9, I10 and I70, for which more than one week of entries are missing.

With the largest data sections of consistent missing data entries established, it is further necessary to identify prolonged power output failures, which represents a fault condition.

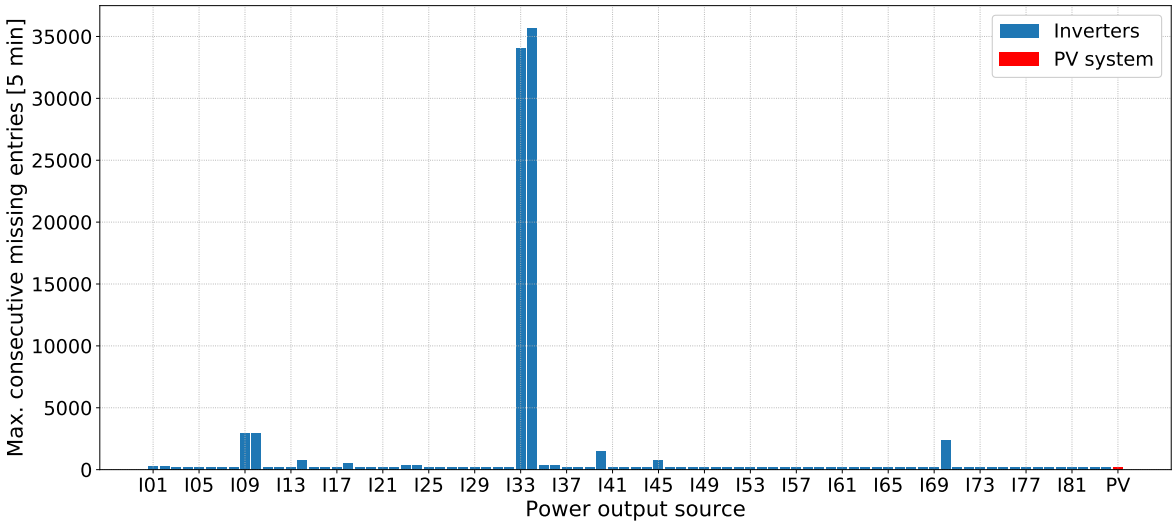


Figure A.28: Total consecutively missing data entries for each power source.

Table A.6: Largest number of consecutive missing data points for the power data sets.

	Inverter number							
	I34	**I33**	**I09**	**I10**	**I70**	**I40**	**I14**	**I45**
Missing entries	35706	34027	2950	2950	2359	1521	796	790
Start date & time	2016/11/23 18:20:00	2016/11/29 14:20:00	2017/10/06 13:35:00	2017/10/06 13:35:00	2017/12/25 08:20:00	2015/11/26 05:45:00	2015/11/23 17:05:00	2017/07/25 17:45:00
End date & time	2017/03/27 17:45:00	2017/03/27 17:50:00	2017/10/16 19:20:00	2017/10/16 19:20:00	2018/01/02 12:50:00	2015/12/01 12:25:00	2015/11/26 11:20:00	2017/07/28 11:30:00

From the data set it is established that when hardware related fault conditions occur, power output is approximately 0 W (usually between 0 W - 3 W due to sensor sensitivity) for an extended period of time and not only during night time. Thus, to identify time segments, where both inverter and macro PV-system power output values of 0 W are consistently recorded during sunshine hours, Figure A.29 illustrates the total number of entries below 3 W (representative of a no power 0 W output condition).

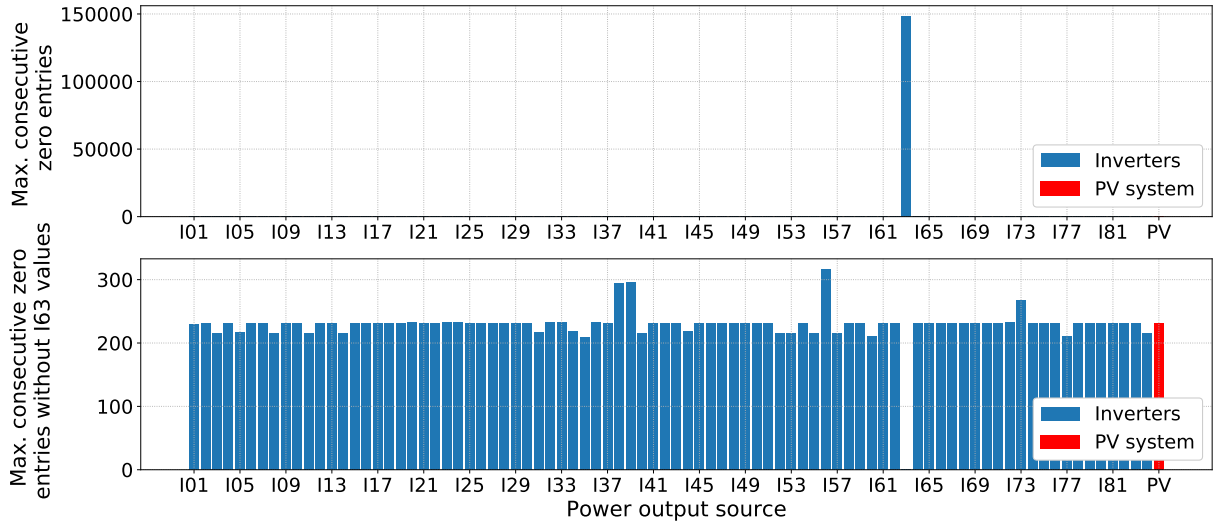


Figure A.29: Top: Total maximum consecutive zero data entries for each power source Bottom: Total consecutive zero data entries counted for each power source without I63.

Evident from the top graph of Figure A.29, is that inverter I63 has the most consistent zero power data entries logged. From analysis it is revealed that zero power output has been logged for the last 17 months of the 4 year data set. To obtain perspective regarding the number of zero power output values measured for the inverters and the macro PV-system power data, the bottom graph of Figure A.29 presents the number of zero data entries, without inverter I63. From this graph, the number of zero power output data points of the inverter and PV-system power data sets are very similar, indicating no further serious data logging issues in this regard.

Inverter data substitution

With the list of inverters identified in Table A.6, together with inverter I63 (17 months of 0 W entries) for which missing data is to be substituted, suitable replacements for each of the large missing inverter data segments must be identified. These suitable replacements for inverter data are determined with a correlation analysis, which identifies the inverter data similarity based on:

A. *Must demonstrate high temporal correlation.* Due to the physical size of the PV-system, different environmental factors are expected to have different effects on PV modules, dependent on the location of these modules within the PV system. For example, if a cloud casts a shadow only on one half of the PV system, then the power output dynamics of the non-shaded inverter modules will be completely different at that moment in time.

B. *Must demonstrate similar power output capacity.* Due to the maximum power output value being influenced by various factors, such as varying rated PV-module capacities, wind circulation, non-uniform dust distribution, soiling, etc., inverters must be highly correlated in terms of the general maximum power output (power amplitude) value.

With this criteria defined, a metric is required to reveal the inverters with a high correlation regarding the time-series power data dynamics. For this purpose, the MSE is identified as the metric of similarity. As presented with Eq. (A.2), variable P_{Inv-i} represents the reference inverter for which possible inverter data replacements P_{Inv-j} are to be identified.

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(P_{Inv-i} - P_{Inv-j})^2 \tag{A.2}$$

Since both power amplitude (criteria B) and out-of-phase time-series differences (criteria A) are penalised with the MSE, this metric is ideal for identifying inverter similarity. Importantly, before the MSE correlation analysis is executed, all missing data entries, as well as known fault conditions are first eliminated from the data set. The power values of each inverter are also normalised relative to itself. By normalising the power data, criteria B regarding power amplitude can be adhered to, without a MSE bias resulting from different PV-module rated capacities, or a difference in the number of PV modules connected to a single inverter (which depends on the configuration).

With the similarity of all 84 inverters determined relative to one another, the MSE correlation matrix with a dimension of 84×84 is established, as presented with Figure A.30. Naturally, from this correlation matrix it is clear that an inverter is most highly correlated with its own power output time-series data set.

By further analysing the MSE correlation graph of Figure A.30, together with the actual inverter locations as indicated in Figure A.31, it is evident that inverters correlate best with adjacent inverters, located to the East and West. The second highest similarity correlation is with adjacent inverters located to the North and South of the reference inverter.

From the correlation analysis the top four suitable inverter replacements (ranked according to correlation similarity) identified for the faulty inverter data entries are summarised in Table A.7.

Table A.7: Suitable inverter replacements for inverters with prolonged faulty data entries.

	Inverter data to be replaced								
	I34	**I33**	**I09**	**I10**	**I70**	**I40**	**I14**	**I45**	**I63**
Best	I36	I35	I11	I12	I74	I42	I16	I47	I61
correlated	I32	I31	I07	I08	I69	I38	I12	I43	I65
inverter	I33	I04	I10	I09	I76	I71	I13	I28	I46
substitutions	I39	I34	I12	I25	I73	I67	I23	I30	I64

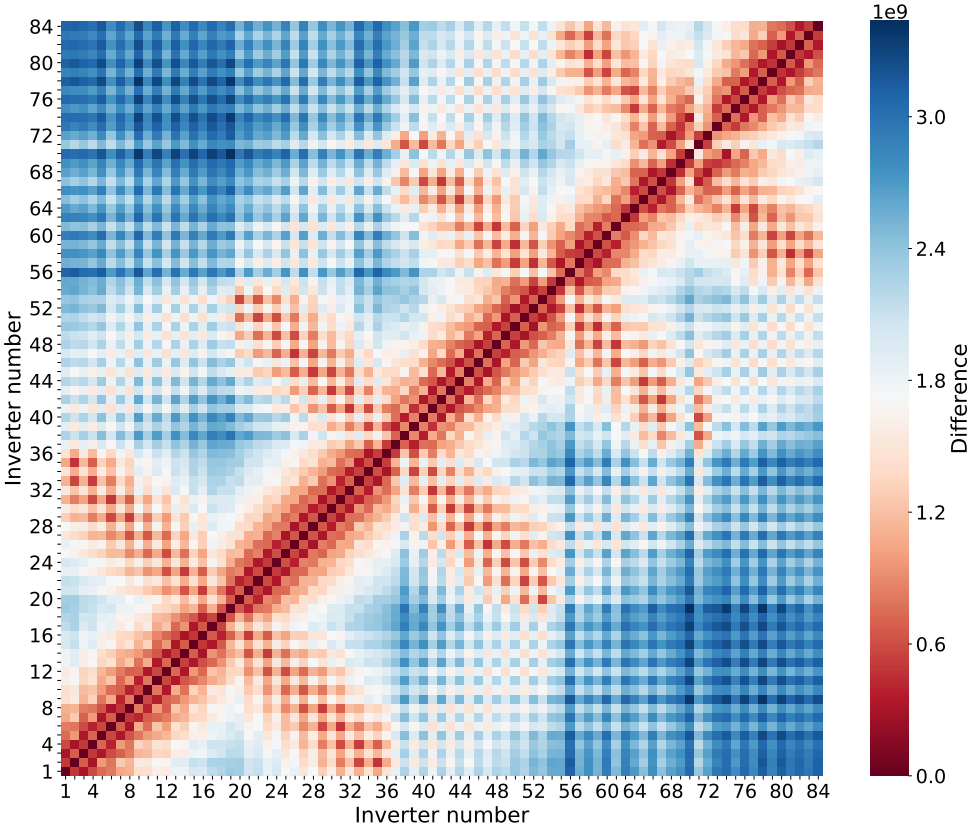


Figure A.30: Inverter MSE correlation matrix, with red indicating high correlation and blue indicating low correlation.

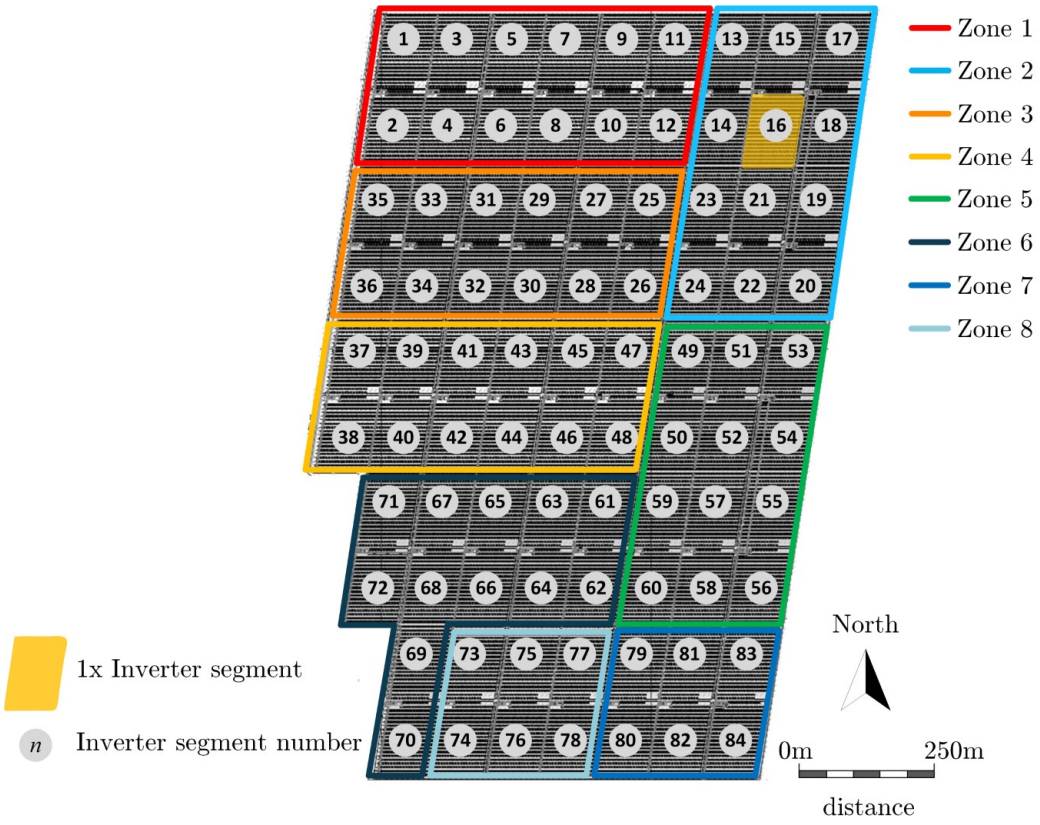


Figure A.31: PV-system layout with inverter positions and zone allocations indicated.

Confirming fault and missing data events

Since the macro PV-system power data is directly proportional to the aggregated inverter power output, this relationship as previously defined with Eq. (A.1), must be accounted for during any changes made to either the inverter or macro power data sets.

Before replacing the large segments of missing/faulty inverter data, it is important to understand that bad data logging and not necessarily a failure of power delivery could be the source of the missing data. Thus, it would be unscientific to simply assume that the inverters did not deliver power for the periods where missing data, or zero power output (fault condition) values are logged.

With this considered, for the replacement of missing and faulty data, it must be established whether or not power was delivered. Power delivery can easily be confirmed from the output of the LV/MV transformers, with each of these transformers (42 in total) receiving power from two inverters. Unfortunately, it is observed that logging issues for the connected inverters typically translate onto the LV/MV data as well. This is suspected to occur, since the same communication line is used for the data of both the inverters and the LV/MV transformer. However, the PV system is electrically divided into eight separate zones, as illustrated in Figure A.31. The collective power output of each zone into the high-voltage transformers is measured separately. Therefore, with this power data available, the relationship of the aggregated inverter power output of each zone can be defined as done with Eq. (A.3):

$$P_{zone-x} = \sum_{i=1}^{M} P_{inv-i} \tag{A.3}$$

Variable P_{inv-i} represents the inverter power delivered by inverter i with P_{zone-x} as the total aggregated power received from all M inverters in the particular zone x. From this power relationship, a scenario where data is correctly logged at both the point of measurement for the individual inverters, as well as the aggregated power output measurement of the inverter zone, should result in a power difference close to zero, as defined by Eq. (A.4):

$$Difference = \sum_{i=1}^{M} P_{inv-i} - P_{zone-x} \approx 0 \tag{A.4}$$

With the date-time segments known for the most consecutive missing and zero-fault entries, inverter data for all of the connected inverters of a particular zone is extracted. To determine whether data is only to be replaced for the inverter data set, or if a data adjustment should also be made for the macro PV-system power data set, Eq. (A.4) is used to determine the action of data adjustments to be made. This action is defined as:

A. *Equation (A.4) True*: This scenario indicates that no power is in fact delivered. Missing or faulty inverter data is imputed with the data from the most highly correlated inverters. Also, the corresponding PV-system power data is updated with the addition of the same data used for the inverter imputation, therefore adhering to the relationship defined with Eq. (A.1). This is demonstrated with Figure A.32.

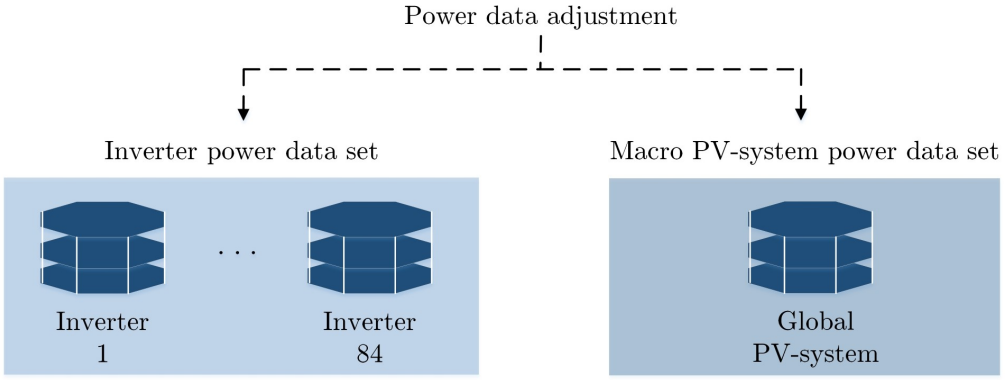


Figure A.32: Power data adjustment made to both inverter and macro PV-system data sets.

B. *Equation (A.4) False*: This scenario indicates a faulty data entry, with power indeed being delivered. Only the missing/faulty inverter data is imputed, with the macro PV-system data set maintained as initially recorded. This is demonstrated with Figure A.33.

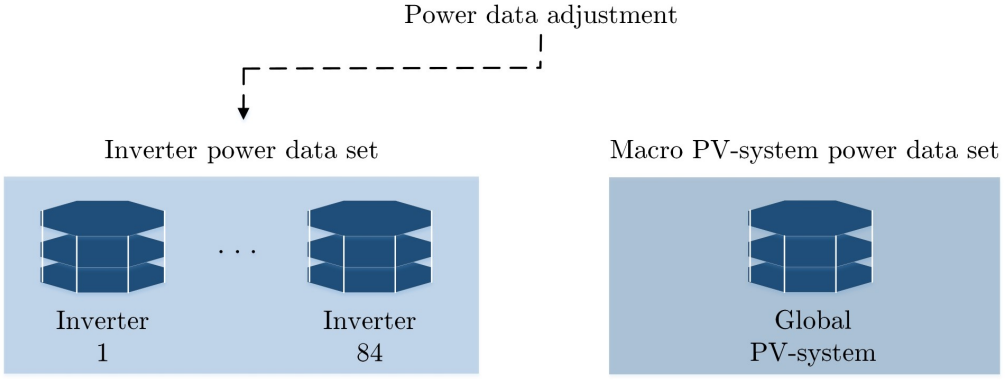


Figure A.33: Power data adjustment made only to inverter data set.

As an example of this procedure, where data is imputed for segments with missing/faulty inverter data, inverters I33, I34 and I63 are analysed. These three inverters present some of the most extreme cases of missing and faulty data logged.

Inverter I33 and I34 data imputation:
Inverters I33 and I34 collectively deliver power together with inverters I25 - I36 (12 inverters total), which are all allocated to Zone-3 as shown in Figure A.31. With the execution of Eq. (A.4), condition A is satisfied. The time period that captures both the time segments when I33 and I34 stopped delivering power, is presented in Figure A.34.

Apart from the 2 MW data spike at 11-12-2016 (attributed to missing Zone-3 measurement data), there is evidently no substantial difference between the aggregated inverter power outputs and the received inverter power as recorded for Zone-3. Should either of the inverters have delivered power, then a continuous value much larger than zero would be visible. Therefore, data is to be substituted for inverters I34 and I33. Importantly, this data is also to be added to the macro PV-system power output data set for this time segment, so that the relationship of Eq. (A.1) is adhered to.

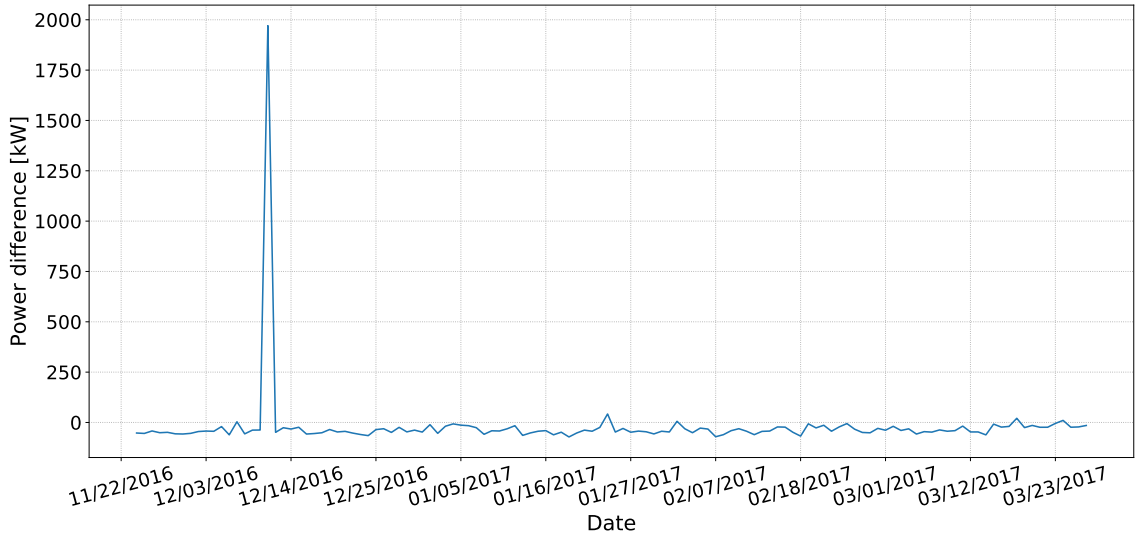


Figure A.34: Average daily power difference between measured aggregated inverter power and the total Zone-3 measurement.

Inverter 63 data imputation:
Inverter I63, together with inverters I61 - I72 (12 inverters total) all collectively deliver power for Zone-6. The recorded data points of I63 for the period of 2018-01-01 to 2019-05-31 are all zero. This could either be attributed to no power being delivered, or to a data logging issue. With the execution of Eq. (A.4), the average daily difference in power recorded for Zone-6 and aggregated power delivered by inverters I61-I72 is displayed in Figure A.35. From this graph it is evident that after the date of 2018-01-01, there is a sudden increase in the difference between total inverter and total zone power data for the remainder of time.

From the graph in Figure A.35 the consistent average daily difference of approximately 400 kW indicates that power is in fact delivered by I63. Therefore, data logging is the issue and only the power data set of inverter I63 is to be adjusted, as determined with condition B satisfied.

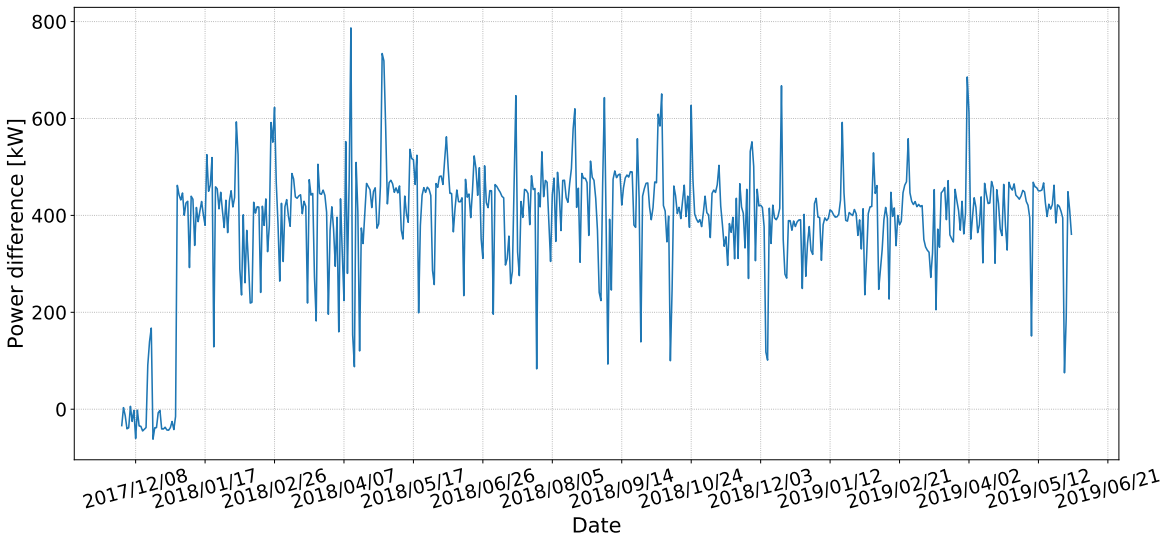


Figure A.35: Average daily power difference between measured aggregated inverter power and the total Zone-6 measurement.

In similar fashion as demonstrated by these examples, further analysis has revealed that the following data adjustments have to be made to the inverters, which are initially flagged in Table A.6 for prolonged periods of missing or zero data:

- I9, I10, I33, I34, I70: Condition A of Eq. (A.4) satisfied. For these data segments inverter and macro PV-system data is to be adjusted accordingly.
- I14, I45, I63: Condition B of Eq. (A.4) satisfied. For these data segments only the inverter data is substituted, but the macro PV-system data set is not adjusted.

Final inverter and PV-system data sets

With most of the large inverter and PV-system data discrepancies identified, data imputation is used to eliminate these data segments and obtain a more complete and representative data set for all inverters and the macro PV-system power output.

Specifically, for the replacement of inverter data, the average of the two most correlated inverters are used as identified from the MSE correlation matrix. For example the missing data of I34 is replaced by the average of the power output data of inverters I36 and I32. This same data used to replace the I34 data is then added to the PV-system power output data, since condition A of Eq. (A.4) is satisfied for this scenario. To illustrate the effect of this data imputation on both the inverter and macro power data sets, the new changes are evident in the time-series graphs of Figure A.36

The final set of inverters utilised to replace the missing/faulty data segments are displayed in Table A.8.

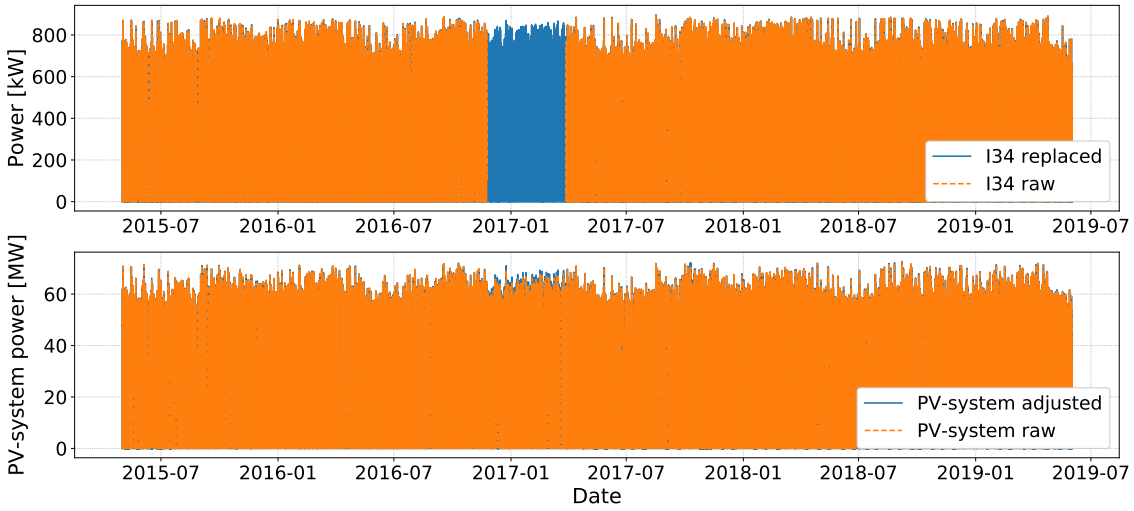


Figure A.36: Top: Inverter I34 imputed data. Bottom: Corresponding adjustment to macro PV-system power time-series data set.

Table A.8: Inverter replacements used for corresponding missing inverter data.

	Inverter data to be replaced								
	I09	**I10**	**I14**	**I33**	**I34**	**I40**	**I45**	**I63**	**I70**
Inverter data replacements	I11 I07	I12 I08	I16 I12	I35 I31	I36 I32	I42 I38	I47 I43	I61 I65	I74 I69

A.4.4 Curtailment elimination

Curtailment refers to a scenario where the power delivered by the PV system is purposefully limited (curtailed). Curtailment is typically initiated in response to demand side management strategies, where for example there is an over supply of solar power, which exceeds the load demand. With regards to the training of the forecast models, days on which curtailment occur are eliminated from the data set. Events of curtailment significantly alter the typical macro-level behaviour of the PV system, which might impair forecast model development. Curtailment is further an external, decision-based variable with no real means of being predicted. Therefore, curtailment is not a variable to be anticipated by the PV forecasting models and is eliminated from the data set.

Days on which curtailment occurred in the data set are identified based on three conditions, which help to avoid the possibility of false database indicators of curtailment:

1. *Active-power-target-value* database variable of inverter must be < 880 kW.
2. More than 8 inverters (representing 10 % of the total number of inverters) must be curtailed simultaneously.
3. Curtailment must be actively applied for more than 15 min during operational hours (5 AM - 8 PM).

Firstly, each inverter has a data point labelled as *Active-power-target-value*. Should this value be less than the rated inverter output power of 880 kW the inverter is flagged for a potential active-curtailment operation. The second condition for the complete confirmation

that the PV system is curtailed, is that at least 10 % of all the inverters (eight inverters of the total 84 inverters) must be in agreement, with the *Active-power-target-value* data point satisfying condition one. The third and final condition is that conditions one and two are satisfied for more than 15 min during the day, between the operational hours defined as 5 AM to 8 PM. This ensures enough redundancy so that curtailment is not wrongfully identified, which will result in the unnecessary elimination of data.

Figure A.37 illustrates all of the instances where curtailment is identified. The total number of days where curtailment occurred is 78 days in total, of which the majority is applied for the extended period of 2015-10-26 to 2015-12-16 (52 days).

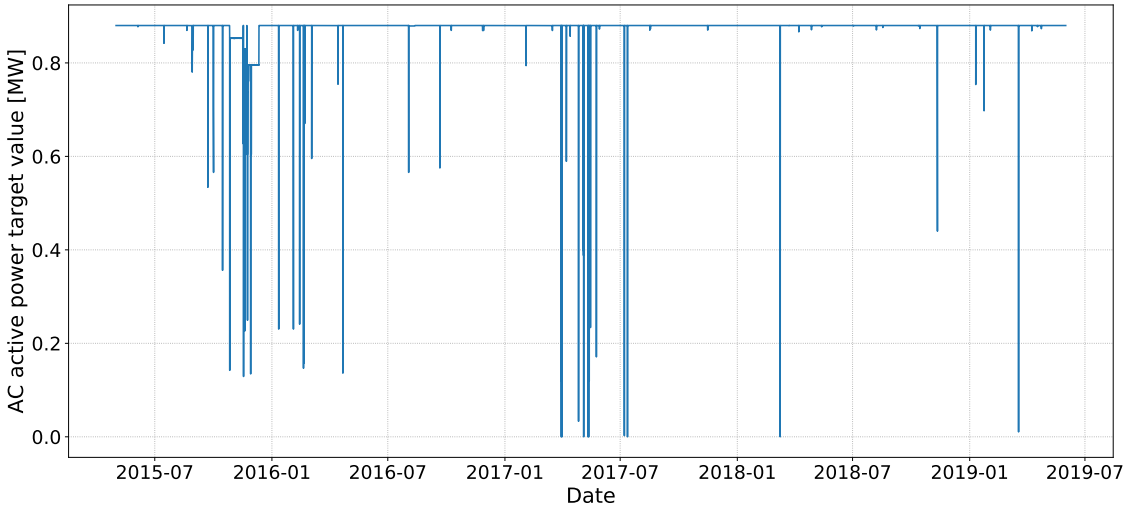


Figure A.37: Occurrences of curtailment during the entire data set period.

As mentioned, due to the inherent seasonality of the data, it is important for the process of model training to be exposed to full cycles of seasonality (1 year). Therefore, all power output data (PV system and inverters) measurements for the extended curtailment period of 2015-10-26 to 2015-12-16 are replaced by the corresponding power data of the following year, for the period of 2016-10-26 to 2016-12-16. This ensures that the process of model training does not result in models being underexposed to the months of November and part of December. This solution is empirically proven to deliver higher forecasting accuracies, as opposed to a scenario where the curtailment data is simply eliminated from the data set, reducing model exposure for the months of November and December. The remaining single days in the data set, for which curtailment is applied are eliminated, had no impact on overall model training.

A.4.5 Data pre-processing concluded

With the three stages of data processing executed as presented in the previous sections of this chapter, the first goal was to obtain a truly representative data set, which accurately captures the PV-system dynamics in response to the environmental and atmospheric variables. This of course requires a data set that is free from anomalies, which could impair forecast model training. Secondly, the goal was to obtain a data set, representative of full cycles of seasonality, so that the forecasting models are unbiased and generalise well in terms of anticipating PV-system behaviour for all weather (daily and seasonal) variations.

With faulty data measurements eliminated and the necessary substitution of data segments made, all of the initial data set issues are illustrated on the time lines of Figure A.38. This graph illustrates the seasonal time-series movement of the received irradiance, to assist with the visualisation of when the data set issues occurred. All of the anomalies have been addressed and dealt with accordingly.

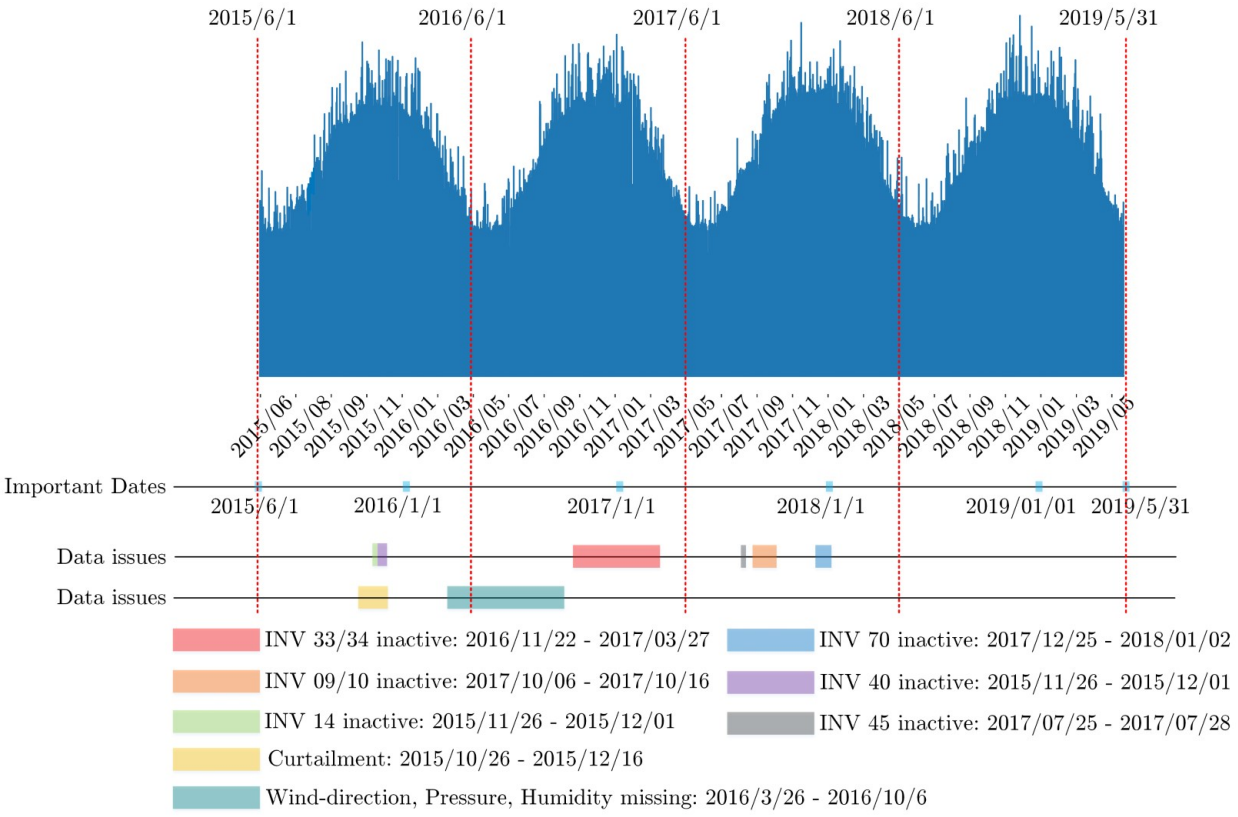


Figure A.38: Summary of the data set irregularities that have been addressed.

This concludes the pre-processing of the received raw data set. The data set up to this point is ready to be used for training the forecast models. Of course, further tailored adjustments of this data set are still to be made, to allow for the execution of the training strategy used for each unique forecasting model. More details regarding this are presented in Chapter 3, which is a discussion of the methodology concerning the process of training the models.

A.5 Correlation analysis

A.5.1 Pearson's correlation method

The Pearson correlation analysis has proven to be a popular choice in PV-forecasting literature [72, 74, 75]. Essentially, this analysis measures the strength of the linear relationship between two variables [122, p. 298]. Equation (A.5) represents the Pearson's correlation coefficient [124]:

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum(x_i - \bar{x})^2)(\sum(y_i - \bar{y})^2)}} \tag{A.5}$$

The variables of Eq. (A.5) represent the following: x_i is the value of x for the i-th observation, y_i is the value of y for the i-th observation. The average of the data set observations x and y are presented with $\bar{x}$ and $\bar{y}$. The value n is the number of observations of each variable and r_{xy} is the Pearson r correlation coefficient of variables x and y. Essentially, the Pearson's correlation coefficient is the covariance of the two variables (x,y) divided by the product of the standard deviation of each sample of data.

The use of the mean ($\bar{x}$,$\bar{y}$) in the calculation suggests that the two variables compared should have a Normal (Gaussian) distribution [122, p. 298]. To avoid any potentially incorrect assumptions regarding the data distributions of the available variables, a rank correlation method is also used, referred to as the Spearman's rank-correlation method.

A.5.2 Spearman's rank-correlation method

Spearman's correlation coefficient ρ measures the strength of association between two ranked variables[124] and is a distribution-free (non-parametric) correlation statistic. Unlike the Pearson's correlation, which strictly measures the strength of linear relationship, it measures the monotonic relationship, where an increase in one variable corresponds with the increase in another. Equation (A.6) presents the formula for calculating ρ as [123]:

$$\rho = 1 - \frac{6\sum_{i=1}^{N} d_i^2}{n(n^2-1)} \tag{A.6}$$

Where the variable d_i of Eq. (A.6) is the difference of the i-th elements of each set of variables. Variable n represents the number of ranks (10 for this research), which have been assigned and finally ρ is the strength of association between the two variables [123].

A.5.3 Application of Pearson and Spearman correlation methods

With Figure A.39 the real-time Pearson and Spearman correlations are demonstrated for each variable, with respect to PV-system power output. Importantly, the absolute values are displayed, since the magnitude of the value is what indicates the strength of association.

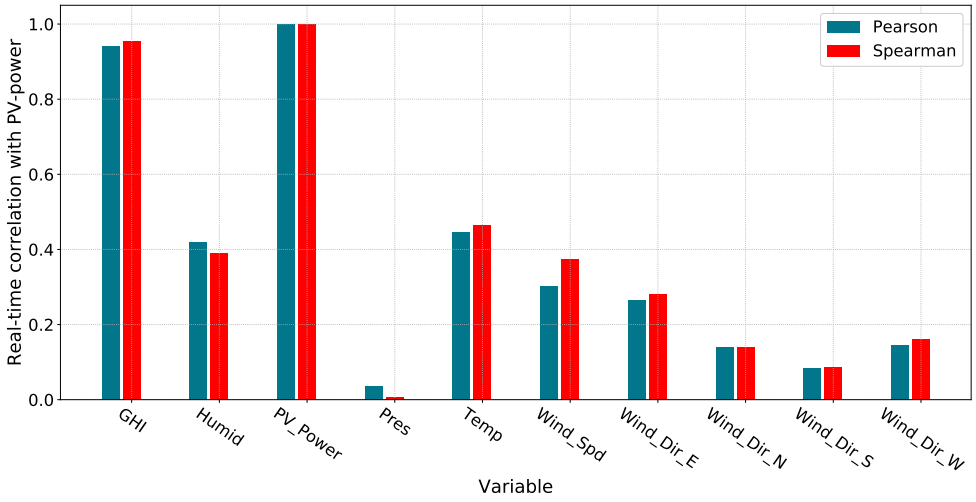


Figure A.39: Real-time correlations of power relative to the various measured data points.

Evident from Figure A.39 is that the Pearson and Spearman correlations are very similar, with both correlation methods allocating the highest correlation to GHI and historic power output, followed by temperature. However, this correlation graph only conveys information regarding the real-time correlations. It is also necessary to investigate the historic time-series correlations of these variables with power output, which is important since models deliver forecasts based on historic observations.

The analysis for this is continued for the Spearman correlation method, where some historic segments of the available input features are correlated with power output. The results of this analysis are presented in Figure A.40. Evident from the correlation results is that GHI and temperature are in general the most well correlated with PV-system power output. Relative humidity and wind speed also have a noticeable correlation. Interestingly, although the wind direction also has a relatively low correlation, there does seem to be a higher degree of influence on power output for wind from the East or West, in comparison to the lower correlation values of the North and South wind directions. The fact that the PV-system modules face North, with rows stretching from East to West allows wind to flow more freely through the rows, which is attributed as a possible reason for the higher correlations observed for East and West wind directions. Wind from the South has the lowest correlation compared to the other wind directions, which is credited to the modules facing North, at a tilt angle of 30°. From the South, wind makes direct contact with the back of the PV-modules and will be guided towards the ground due to the tilt, which restricts the flow of wind over the rest of the PV-system.

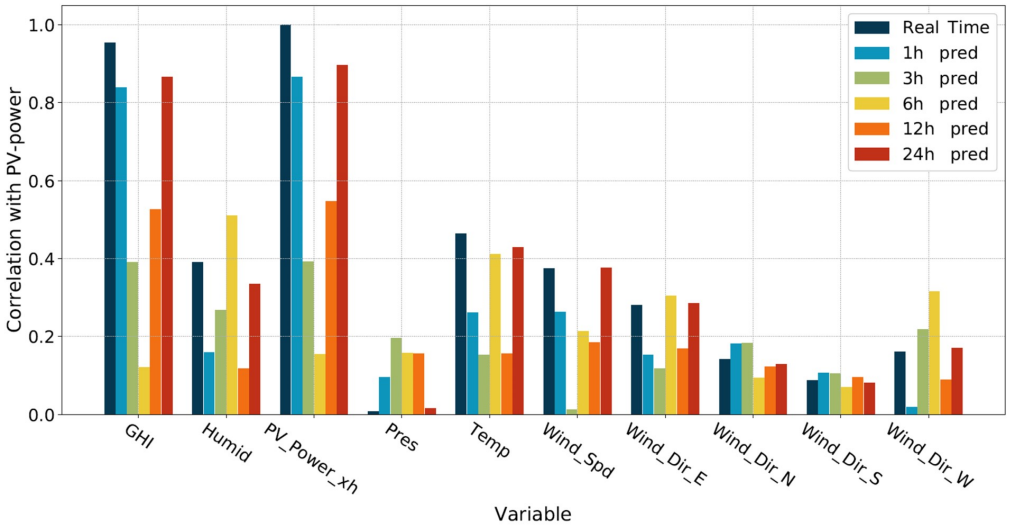


Figure A.40: Spearman correlation analysis between historic (lagged observations) data and real-time PV-system power output.

From the correlation results of Figure A.40, it is evident that real-time and 24 h historic data are very similar regarding the correlations. As expected, the relevance of the historic GHI and previous PV-system power output (PV_Power_xh) values are in agreement, since there is a very strong correlation between power and GHI in general. An interesting observation is the historic value of humidity, which seems to have a high correlation at especially a 6 h historic period. Pressure, although still very low i.t.o. correlation, also seems to have more value for the 1 - 12 h historic data sets, showing less correlation at real-time. Temperature data captured at real-time, 6 h and 24 h evidently also has a good correlation with real-time power output.

Appendix B

Earth Sun movement

B.1 Sun position relative to a specific location

Sun movement, relative to a specific location on Earth, requires three values. These are the solar azimuth (Φ_S) and altitude (β) angles,as well as the distance between the earth and the sun [140, p.29]. The orbit of the Earth around the sun is elliptical, as illustrated Figure B.1. However, due what is referred to as Milankovitch Oscillations, the orbital shape of the Earth changes from elliptical to circular and vice versa approximately every 100,000 years [95, pp.192-193]. However, due to the slow period of transition of these orbital variations, the distance to the Sun can be assumed constant.

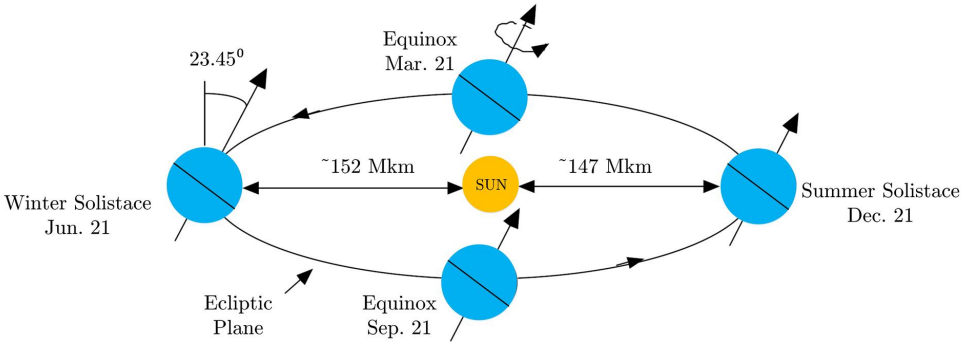


Figure B.1: Illustration of the ecliptic movement around the Sun with Earth at a tilt angle. *Source:* Redrawn illustration as presented by G. Masters [95, p.192]

To determine the (Φ_S) and (β) angles, the value of the declination angle δ [deg] is necessary, which is the angle formed between the centre of the Earth and the centre of the Sun, as illustrated in Figure B.2 [95, p.193]. The deviation of the Sun, as measured from directly above the equator, is effectively what δ represents [140, p.27]. With the plane of orbit as reference, the equator of Earth is tilted at 23.45°, with Eq. (B.1) representing δ at any given day (n) of the year [95, p.194]

$$\delta = 23.45 \left[\frac{360}{365}(n - 81)\right] \tag{B.1}$$

With Figure B.3 an illustration is provided regarding how the β and Φ_S angles give context of the movement of the Sun relative to a specific point on Earth. The angular measure of

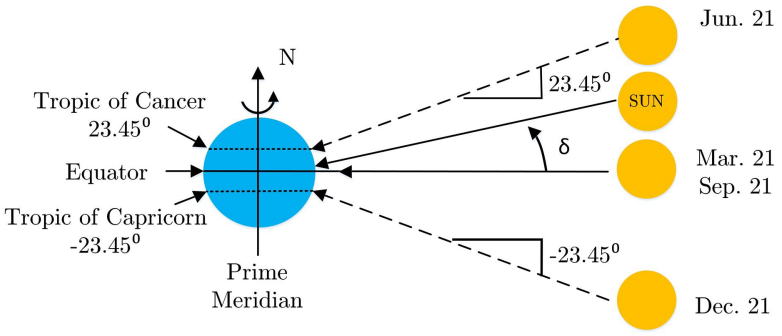


Figure B.2: Illustration of the resulting Declination angle as a result of the Earth and Sun movement. *Source:* Redrawn illustration as presented by G. Masters [95, p.193].

altitude from the local horizon of Earth towards the geometric centre of the Sun is presented by β [94]. The position of the Sun, East or West of North as reference line (analysed 75 MW PV-system located in the Southern hemisphere) is described by Φ_S. Therefore, following the sign convention of the diagram, the Φ_S angle is negative in the afternoon, when the Sun is West of North, and positive in the morning when Φ_S is East of North [95, p.196-197]. At solar noon, with the sun rays aligned perpendicular to a given line of longitude [94], Φ_S is 0°. The angular distance to the West or East of the Prime Meridian, situated at 0°, is represented by the longitude angle. The angular distance North or South of the equator is represented by the Latitude angle. Therefore, for locations on Earth along a mutual line of longitude, solar noon always occurs at the same time. The fact that the PV-system analysed is located in South Africa, which is below the Tropic of Capricorn, means that solar noon is experienced to the North of the PV modules.

With the future position of the Sun available from the β and Φ_S angles, the forecast models are given a valuable set of input data towards predicting PV-system power output (highly correlated with sun movement).

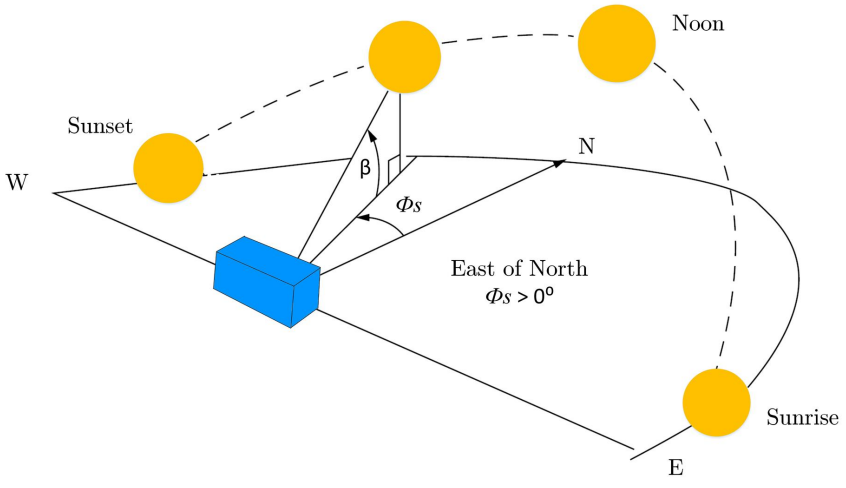


Figure B.3: Illustration of Sun position determined by altitude angle β and azimuth angle Φ_S. *Source:* Redrawn illustration as presented by G. Masters [95, p.197].

The β and Φ_s angles are respectively determined by equations B.2 and B.3 [95, p.197]:

$$\beta = \sin^{-1}(\cos L \cos\delta \cos H + \sin L \; sin\delta) \tag{B.2}$$

$$\Phi_S = \sin^{-1}\left(\frac{\cos\delta \sin H}{\cos\beta}\right) \tag{B.3}$$

The magnitude of Φ_S can however be greater than 90°, should the Azimuth angle Φ_S be more than 90° away from North [95, p198]. Thus, it is necessary to confirm the magnitude of Φ_S with the use of Eq. (B.4) [95, p.198]:

$$\text{if } \cos H \geq \frac{\tan\delta}{\tan L} \text{ then } |\Phi_S| \leq 90° \text{ else } |\Phi_S| > 90° \tag{B.4}$$

Where the variables in B.2 - B.4 are defined as:

L = Latitude [deg]
H = Hour angle [deg]
δ = Declination angle [deg]

Equation (B.5) defines the hour angle H [deg], of which the value is the angular representation of time. For the hours before and after solar noon, the value of H is positive and negative, respectively [95, p. 199]. With Eq. (B.6), the value regarding the $Hours\ Before\ Solar\ Noon$ [hours] is determined.

$$H = \left(\frac{15°}{Hour}\right) \times (Hours\ Before\ Solar\ Noon) \tag{B.5}$$

$$Hours\ Before\ Solar\ Noon = 12:00 - ST \tag{B.6}$$

The variable ST refers to the solar time, which is based on the physical position of the Sun relative to the longitude angle, which serves as the point where solar noon occurs. Clock/Civil Time (CT) on the other hand provides the value of time relative to the longitudinal line used as global reference by the people of a society/country. With the aim to obtain the Φ and β angles as input features for the ML models, it is necessary to convert the solar time into civil time, since forecasts are to be delivered with reference to civil time. The conversion between Clock/Civil Time (CT) [min] and Solar Time (ST) [min] is executed with the use of the following formulas [95, p.207]:

$$ST = CT + \frac{4[\text{min}]}{1°}(Local\ Time\ Meridian - Local\ Longitude)° + EOT \tag{B.7}$$

$$EOT = 9.87\sin 2B - 7.53\cos B - 1.5\sin B \tag{B.8}$$

$$B = \frac{360}{364}(n - 81) \tag{B.9}$$

Where variables n refer to the day number and EOT refers to the Equation of Time [min]. Equation (B.8) is the longitude correction between CT and ST. The second correction term is variable B [deg] as represented by Eq. (B.9), which accounts for the varying length of

solar days due to the elliptical orbit of Earth [95, pp.206,207].

As presented by I. Reda and A. Andreas [141], there are more sophisticated and accurate solar-position algorithms, as determined from the use of Judian calendar days and etc. However, the errors introduced by the formulas presented above with Eq. (B.1) - (B.9) are negligible.

Appendix C

Macro-level model development

C.1 FFNN

With the final selection of the best FFNN PV-system model hyperparameters for all weather conditions, the final stage of model optimisation is done regarding various MB sizes. These results are presented in Table C.1. As evident from the final stage of hyperparameter optimisation, the best MB size for the given architecture, Adam optimiser and LR of $1e^{-4}$ is 64.

Table C.1: Best FFNN model architecture results as combined for various MB sizes.

sliding window	**hidden layers**	**hidden units**	**mini-batch**	**NRMSE [%]**
1 h	3	64	4	9.68
			8	9.67
			16	8.29
			32	8.27
			64	**8.24**
			128	8.27
			256	8.26
Input features	P, GHI, T			
Future input features	month of year, solar angles Φ_S, β			

C.2 LSTM-RNN

With the final selection of the best LSTM-RNN PV-system model architecture for all weather conditions, the final stage of model optimisation is done regarding various MB sizes. These results are presented in Table C.2. As evident from the final stage of hyperparameter optimisation, the best MB size for the given architecture is 32.

Table C.2: Best LSTM-RNN model architecture results as combined for various MB sizes.

sliding window	hidden layers	hidden units	mini-batch	NRMSE [%]
3 h	3	16	4	8.55
			8	8.44
			16	8.48
			32	**8.36**
			64	8.51
			128	8.50
			256	8.58
Input features	P, GHI, T			
Future input features	month of year, solar angles Φ_S, β			

C.3 GRU-RNN

With the final selection of the best GRU-RNN PV-system model architecture for all weather conditions, the final stage of model optimisation is done regarding various MB sizes. These results are presented in Table C.3. As evident from the final stage of hyperparameter optimisation, the best MB size for the given architecture is 64.

Table C.3: Best GRU-RNN model architecture results as combined for various MB sizes.

sliding window	hidden layers	hidden units	mini-batch	NRMSE [%]
6 h	2	64	4	8.27
			8	8.25
			16	8.22
			32	8.40
			64	**8.16**
			128	8.19
			256	8.18
Input features	P, GHI, T, AP			
Future input features	month of year, solar angles Φ_S, β			

C.4 Macro-level forecast model time-series graphs

The graphs in Figures C.1 - C.4 illustrate the time-series forecasting ability of the three macro-level forecasting models developed, which are the FFNN, LSTM-RNN and GRU-RNN models. The graphs illustrate the 1 h, 3 h and 6 h forecast time steps, since this provides a good general context of model performance as the forecast horison increases and forecasting difficulty progressively increases as well. Subsequently, an increase in intermittence of received irradiance further increases forecasting difficulty.

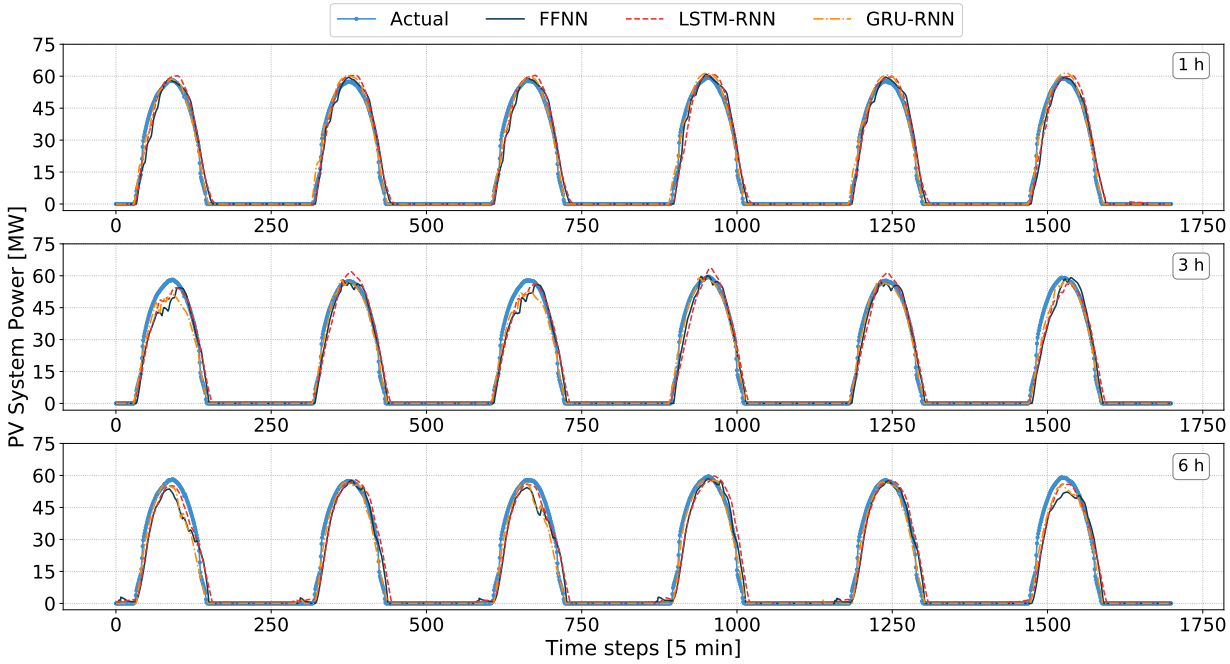


Figure C.1: Time-series forecasts delivered 1 h, 3 h and 6 h ahead for clear days.

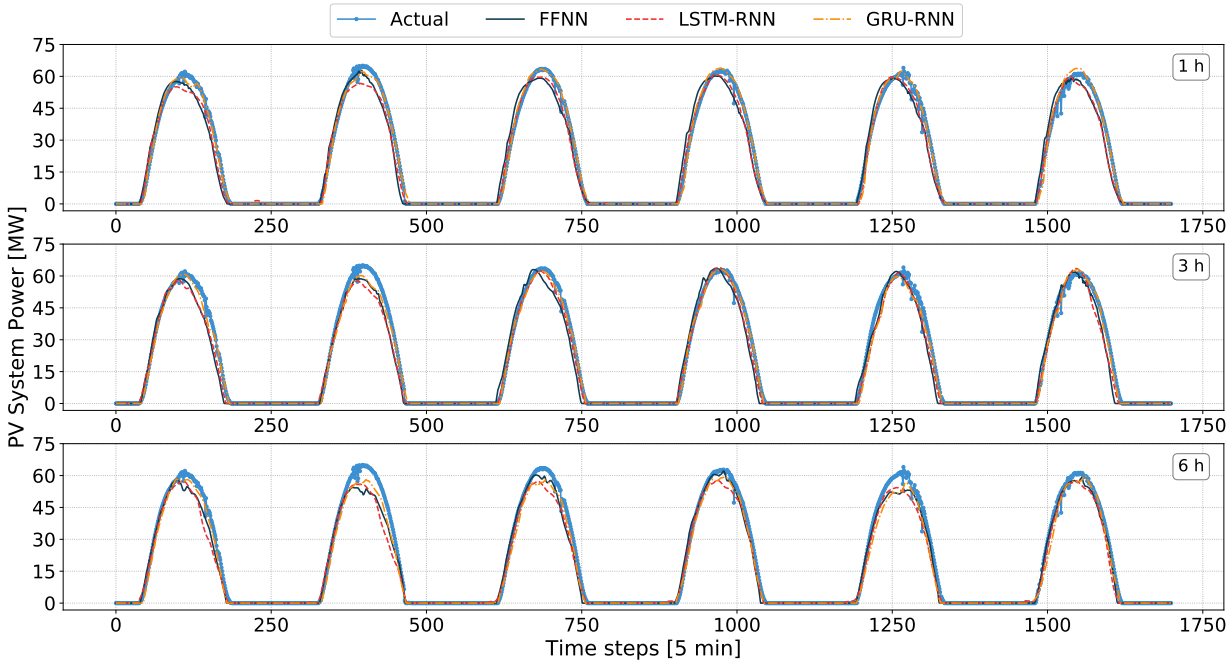


Figure C.2: Time-series forecasts delivered 1 h, 3 h and 6 h ahead for clear-intermittent days.

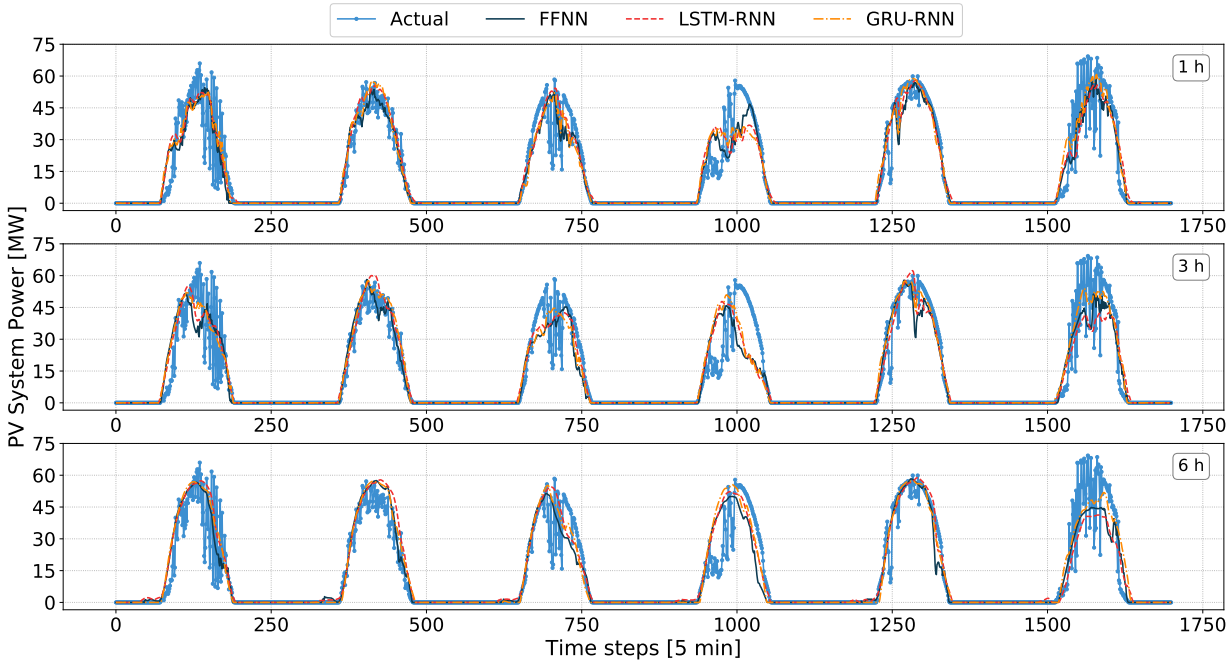


Figure C.3: Time-series forecasts delivered for 1 h, 3 h and 6 h ahead for intermittent days.

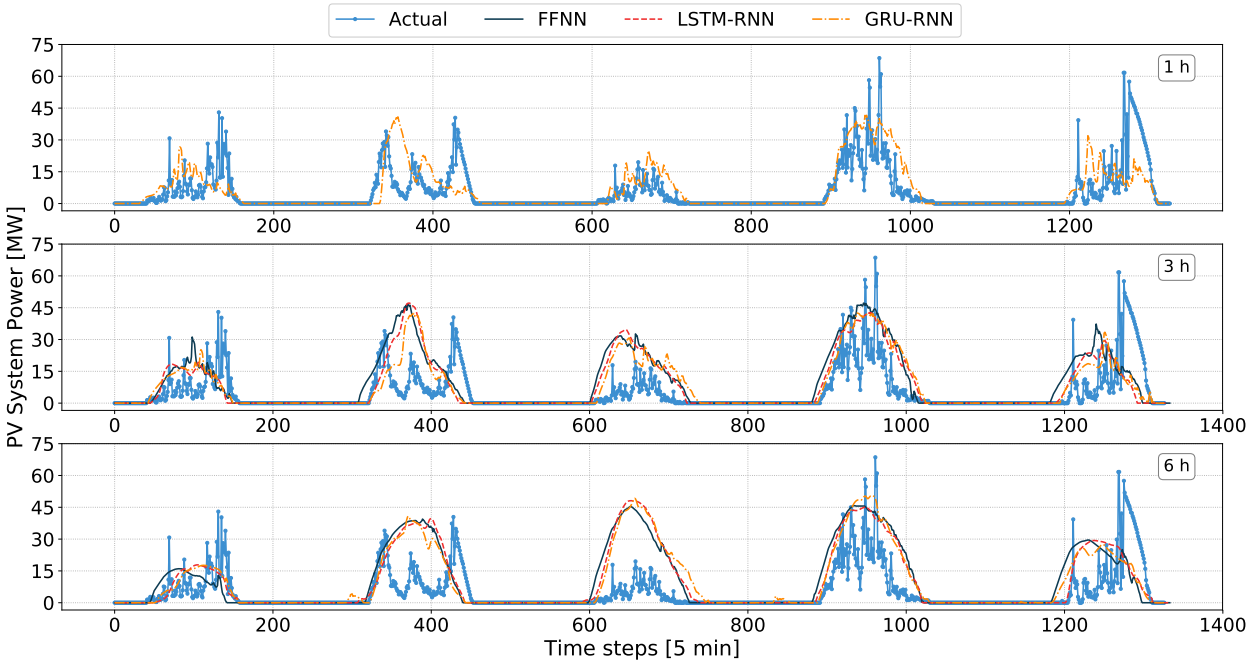


Figure C.4: Time-series forecasts delivered for 1 h, 3 h and 6 h ahead for overcast days.

Appendix D

Inverter-level model development

D.1 FFNN sliding-window preference

Given sets of different input features, an analysis is performed to investigate the best sliding-window size for inverters at different locations within the PV system. Three inverters are specifically chosen, representative of PV-module sections, which are anticipated to have a different exposure and therefore response to the immediate environment. From the results in Table D.1, the 1 h sliding-window size evidently delivers the best results for the P, GHI and T input features.

The same analysis is also performed for the input feature set consisting of variables P, GHI, T, WS and WD. Once again, the same sliding-window size of 1 h has delivered the lowest NRMSE value.

Table D.1: Sliding-window size results for the FFNN aggregated inverter-level model with historic input features: P, GHI, T

	NRMSE [%]		
sliding window	**INV-10**	**INV-53**	**INV-67**
1 h	**9.218**	**9.205**	**9.198**
2 h	9.219	9.213	9.213
3 h	9.249	9.205	9.213
6 h	9.236	9.229	9.234
24 h	9.307	9.330	9.306
Historic input features	P,GHI,T		
Future input features	month of year, solar angles Φ_S, β		

Table D.2: Sliding-window size results for the FFNN aggregated inverter-level model with historic input features: P, GHI, T, WS, WD

	NRMSE [%]		
sliding window	**INV-10**	**INV-53**	**INV-67**
1 h	**9.392**	**9.359**	**9.371**
2 h	9.414	9.366	9.419
3 h	9.508	9.472	9.494
6 h	9.593	9.508	9.582
24 h	9.852	9.896	9.795
Historic input features	P,GHI,T,WS,WD		
Future input features	month of year, solar angles Φ_S, β		

D.2 FFNN inverter-cluster models: Models *A* and *B*

Table D.3 and D.4 present the FFNN inverter-level model hyperparameter selections, as was determined for the 10 representative inverters of each cluster with the completion of Phase-3 model development. The Rank Choice column indicates the top two model hyperparameter combinations as identified for the specific representative inverter.

Table D.3: FFNN *A* inverter-level model hyperparameter selections.

Representative inverter	Rank Choice	Hidden Layers	Hidden Units	Mini-batch size
4	1st	2	128	64
	2nd	2	512	32
10	1st	2	128	16
	2nd	3	64	256
16	1st	2	128	64
	2nd	3	64	16
39	1st	2	512	128
	2nd	3	64	32
47	1st	3	64	16
	2nd	2	512	32
53	1st	3	64	128
	2nd	2	128	16
67	1st	3	64	32
	2nd	2	512	128
59	1st	3	64	32
	2nd	2	128	64
74	1st	3	64	64
	2nd	2	128	128
81	1st	2	128	16
	2nd	3	64	256

Table D.4: FFNN *B* inverter-level model hyperparameter selections.

Representative inverter	Rank Choice	Hidden Layers	Hidden Units	Mini-batch size
4	1st	2	128	64
	2nd	2	512	64
10	1st	2	128	64
	2nd	2	1024	256
16	1st	2	1024	256
	2nd	3	1024	32
39	1st	2	128	64
	2nd	2	1024	256
47	1st	2	128	64
	2nd	2	1024	256
53	1st	3	128	64
	2nd	2	1024	256
67	1st	2	128	64
	2nd	2	1024	256
59	1st	2	1024	256
	2nd	2	128	64
74	1st	2	1024	256
	2nd	3	1024	32
81	1st	2	128	64
	2nd	2	512	128

D.3 GRU-RNN sliding window preference

Given sets of different input features, an analysis is performed to investigate the best sliding-window size for inverters at different locations within the PV system. Three inverters are specifically chosen, representative of PV-module sections, which are anticipated to have a different exposure and therefore response to the immediate environment. With the results of Table D.5, the 6 h sliding-window size evidently delivers the best accuracy for the P, GHI and T input features.

The same analysis is also performed for the P, GHI and T input features, with the addition of AP, WS and WD included for various iterations, as summarised in Tables D.6 - D.8. Once again, for all of these iterations, the same sliding-window size of 6 h has delivered the lowest NRMSE values, as determined for the validation data set.

In terms of the 24 h sliding-window, these iterations were ruled out as feasible solutions, due to very high computational expense and in some cases GPU memory issues regarding training.

Table D.5: Sliding-window size results for the GRU-RNN aggregated inverter-level model with historic input features: P, GHI, T

	NRMSE [%]		
sliding window	**INV-10**	**INV-53**	**INV-67**
1 h	9.24	9.22	9.22
2 h	9.23	9.19	9.21
3 h	9.19	9.16	9.15
6 h	**9.13**	**9.12**	**9.14**
24 h	Not Feasible	Not Feasible	Not Feasible
Historic input features	P,GHI,T		
Future input features	month of year, solar angles Φ_S, β		

Table D.6: Sliding-window size results for the GRU-RNN aggregated inverter-level model with historic input features: P, GHI, T, AP

	NRMSE [%]		
sliding window	**INV-10**	**INV-53**	**INV-67**
1 h	9.24	9.22	9.22
2 h	9.23	9.20	9.20
3 h	9.21	9.18	9.16
6 h	**9.17**	**9.14**	**9.12**
24 h	Not Feasible	Not Feasible	Not Feasible
Historic input features	P,GHI,T,AP		
Future input features	month of year, solar angles Φ_S, β		

Table D.7: Sliding-window size results for the GRU-RNN aggregated inverter-level model with historic input features: P, GHI, T, WS, WD

	NRMSE [%]		
sliding window	**INV-10**	**INV-53**	**INV-67**
1 h	9.35	9.36	9.35
2 h	9.31	9.28	9.31
3 h	9.32	9.31	9.34
6 h	**9.24**	**9.22**	**9.23**
24 h	Not Feasible	Not Feasible	Not Feasible
Historic input features	P,GHI,T,WS,WD		
Future input features	month of year, solar angles Φ_S, β		

Table D.8: Sliding-window size results for the GRU-RNN aggregated inverter-level model with historic input features: P, GHI, T, WS, WD, AP

	NRMSE [%]		
sliding window	**INV-10**	**INV-53**	**INV-67**
1 h	9.37	9.36	9.37
2 h	9.32	9.31	9.30
3 h	9.32	9.29	9.35
6 h	**9.18**	**9.23**	**9.20**
24 h	Not Feasible	Not Feasible	Not Feasible
Historic input features	P,GHI,T,WS,WD,AP		
Future input features	month of year, solar angles Φ_S, β		

D.4 GRU-RNN inverter-cluster models: Models *C* and *D*

The architectures identified for each of the representative inverters, which deliver the highest forecasting accuracies for Phase-1 of model development are presented in Table D.9 and Table D.10. With the top two architectures extracted for each representative inverter, several edge cases are identified for the initial grid-search space, as highlighted in blue. These edge case hyperparameter solutions are further extended according to the guided grid-search development strategy (discussed in Section 3.6.2), which results in the HL size extended to HL=4 from HL=3. Inverter models with an edge case for HL=3 and HUs=16 are further extended with the HUs ranging between 8, 16 and 32 combined with a HL=4 selection. On the other hand, edge-case solutions for HL=3 and HUs=64 are extended for HL=4 and HUs ranging between 32, 64 and 128 per HL (constant HL width maintained). Inverter models with an edge case for HL=2 and HUs=16 are further extended with HUs=8. Similar to Phase-1, these iterations are completed for both MB sizes of 32 and 64.

Table D.9: GRU-RNN inverter-level model *C* Phase-1 best HL and HU combinations.

	1st Choice		2nd Choice	
Inverter	**Hidden Layers**	**Hidden Units**	**Hidden Layers**	**Hidden Units**
I4	2	[16, 16]	2	[32, 32]
I10	2	[32, 32]	2	[16, 16]
I16	3	[16, 16, 16]	2	[16, 16]
I39	3	[16, 16, 16]	2	[16, 16]
I47	2	[32, 32]	3	[16, 16, 16]
I53	2	[32, 32]	3	[16, 16, 16]
I67	2	[32, 32]	3	[256, 256, 256]
I59	2	[16, 16]	3	[16, 16, 16]
I74	3	[16, 16, 16]	2	[16, 16]
I81	2	[32, 32]	2	[16, 16]
	edge case			

Table D.10: GRU-RNN inverter-level model *D* Phase-1 best HL and HU combinations.

	1st Choice		2nd Choice	
Inverter	**Hidden Layers**	**Hidden Units**	**Hidden Layers**	**Hidden Units**
I4	2	[32, 32]	2	[64, 64]
I10	3	[16, 16, 16]	2	[64, 64]
I16	2	[32, 32]	3	[16, 16, 16]
I39	2	[64, 64]	3	[16, 16, 16]
I47	3	[64, 64, 64]	3	[16, 16, 16]
I53	3	[16, 16, 16]	3	[64, 64, 64]
I67	2	[64, 64]	3	[16, 16, 16]
I59	3	[16, 16, 16]	3	[64, 64, 64]
I74	2	[32, 32]	3	[16, 16, 16]
I81	2	[64, 64]	3	[16, 16, 16]

edge case

From the new hyperparameter solution spaces explored for the appropriate representative inverter models, the best architectures identified are displayed in Table D.11 and Table D.12. Once again several edge cases are identified where models prefer a HL=4 selection. Continuing with the guided grid-search, the hyperparameter solution space is further expanded towards HL=5. However, with a higher computational expense for so many HLs and no further improvements in accuracy, Phase-2 is completed for the inverter-level GRU-RNN models.

Table D.11: GRU-RNN inverter-level model *C* Phase-2 best HL and HU combinations.

	1st Choice		2nd Choice	
Inverter	**Hidden Layers**	**Hidden Units**	**Hidden Layers**	**Hidden Units**
I4	2	[16, 16]	2	[32, 32]
I10	2	[32, 32]	2	[16, 16]
I16	3	[16, 16, 16]	2	[16, 16]
I39	3	[16, 16, 16]	2	[16, 16]
I47	2	[32, 32]	3	[16, 16, 16]
I53	2	[32, 32]	4	[8, 8, 8, 8]
I67	2	[32, 32]	4	[512, 512, 512, 512]
I59	2	[16, 16]	3	[16, 16, 16]
I74	3	[16, 16, 16]	2	[16, 16]
I81	2	[32, 32]	2	[16, 16]

edge case

Table D.12: GRU-RNN inverter-level model *D* Phase-2 best HL and HU combinations.

	1st Choice		2nd Choice	
Inverter	**Hidden Layers**	**Hidden Units**	**Hidden Layers**	**Hidden Units**
I4	2	[16, 16]	2	[64, 64]
I10	3	[16, 16, 16]	2	[64, 64]
I16	2	[32, 32]	3	[16, 16, 16]
I39	2	[64, 64]	4	[16, 16, 16, 16]
I47	3	[64, 64, 64]	4	[16, 16, 16, 16]
I53	3	[16, 16, 16]	4	[32, 32, 32, 32]
I67	2	[64, 64]	4	[16, 16, 16, 16]
I59	3	[16, 16, 16]	3	[64, 64, 64]
I74	2	[32, 32]	4	[32, 32, 32, 32]
I81	2	[64, 64]	3	[16, 16, 16]
	edge case			

Table D.13 and Table D.14 presents the GRU-RNN inverter-level model hyperparameter selections, as was determined with Phase-3 for the 10 representative inverters of each cluster. The Rank Choice column indicates the top two model hyperparameter combinations as identified for the specific representative inverter.

Table D.13: GRU-RNN inverter-level model *C* hyperparameter selections.

Representative inverter	**Rank Choice**	**Hidden Layers**	**Hidden Units**	**Mini-batch size**
4	1st	2	16	32
	2nd	2	32	32
10	1st	2	32	32
	2nd	2	16	64
16	1st	3	16	64
	2nd	2	16	64
39	1st	3	16	32
	2nd	2	16	64
47	1st	2	32	32
	2nd	3	16	32
53	1st	2	32	32
	2nd	4	8	64
67	1st	2	32	32
	2nd	4	512	64
59	1st	2	16	32
	2nd	3	16	128
74	1st	3	16	32
	2nd	2	16	32
81	1st	2	32	32
	2nd	2	16	128

Table D.14: GRU-RNN inverter-level model *D* hyperparameter selections.

Representative inverter	Rank Choice	Hidden Layers	Hidden Units	Mini-batch size
4	1st	2	16	128
	2nd	2	64	128
10	1st	3	16	32
	2nd	2	64	32
16	1st	2	32	64
	2nd	3	16	32
39	1st	2	64	64
	2nd	4	16	64
47	1st	3	64	32
	2nd	4	16	32
53	1st	3	16	32
	2nd	4	32	64
67	1st	2	64	64
	2nd	4	16	64
59	1st	3	16	64
	2nd	3	64	64
74	1st	2	32	32
	2nd	4	32	64
81	1st	2	64	64
	2nd	3	16	64

D.5 GRU-RNN inverter-cluster models: Models *E* and *F*

This section provides the prelimenary results obtained for the addition of WS and WD to potentially improve inverter-level forecasts. The GRU-RNN inverter-level model input feature sets are defined and labelled as:

- *E*: Aggregated inverter-level GRU-RNN with input features: P, GHI, T, WS, WD
- *F*: Aggregated inverter-level GRU-RNN with input features: P, GHI, T, AP, WS, WD

With Phase-1 of model development completed, the following architectures, which delivered the best results for the aggregated inverter-level models *E* and *F*, are summarised in Table D.15 and Table D.16, respectively.

Table D.15: GRU-RNN inverter-level model *E* Phase-1 best HL and HU combinations.

	1st Choice		2nd Choice	
Inverter	**Hidden Layers**	**Hidden Units**	**Hidden Layers**	**Hidden Units**
I4	2	[32, 32]	3	[16, 16, 16]
I10	2	[128, 128]	2	[32, 32]
I16	2	[32, 32]	2	[64, 64]
I39	3	[128, 128, 128]	2	[32, 32]
I47	3	[16, 16, 16]	2	[32, 32]
I53	2	[64, 64]	3	[16, 16, 16]
I67	3	[16, 16, 16]	3	[32, 32, 32]
I59	3	[16, 16, 16]	2	[64, 64]
I74	3	[128, 128, 128]	2	[32, 32]
I81	2	[32, 32]	3	[16, 16, 16]

edge case

Table D.16: GRU-RNN inverter-level model *F* Phase-1 best HL and HU combinations.

	1st Choice		2nd Choice	
Inverter	**Hidden Layers**	**Hidden Units**	**Hidden Layers**	**Hidden Units**
I4	2	[32, 32]	2	[128, 128]
I10	2	[32, 32]	3	[64, 64, 64]
I16	2	[32, 32]	3	[32, 32, 32]
I39	2	[32, 32]	3	[64, 64, 64]
I47	3	[64, 64, 64]	2	[32, 32]
I53	2	[32, 32]	2	[128, 128]
I67	2	[32, 32]	3	[64, 64, 64]
I59	2	[128, 128]	2	[32, 32]
I74	3	[256, 256, 256]	2	[128, 128]
I81	2	[32, 32]	2	[128, 128]

edge case

With Phase-2 and Phase-3 of model development executed, the final hyperparameter allocations are established for each of the 10 representative inverters. The hyperparameter allocations are summarised in Table D.17 and Table D.18 for aggregated inverter-level models *E* and *F*, respectively.

Table D.17: GRU-RNN inverter-level model *E* hyperparameter selections.

Representative inverter	Rank Choice	Hidden Layers	Hidden Units	Mini-batch size
4	1st	2	[16, 16]	32
	2nd	2	[32, 32]	32
10	1st	2	[32, 32]	32
	2nd	2	[16, 16]	64
16	1st	3	[16, 16, 16]	64
	2nd	2	[16, 16]	64
39	1st	3	[16, 16, 16]	32
	2nd	2	[16, 16]	64
47	1st	2	[32, 32]	32
	2nd	3	[16, 16, 16]	32
53	1st	2	[32, 32]	32
	2nd	4	[8, 8, 8, 8]	64
67	1st	2	[32, 32]	32
	2nd	4	[512, 512, 512, 512]	64
59	1st	2	[16, 16]	32
	2nd	3	[16, 16, 16]	128
74	1st	3	[16, 16, 16]	32
	2nd	2	[16, 16]	32
81	1st	2	[32, 32]	32
	2nd	2	[16, 16]	128

Table D.18: GRU-RNN inverter-level model *F* hyperparameter selections.

Representative inverter	Rank Choice	Hidden Layers	Hidden Units	Mini-batch size
4	1st	2	[32, 32]	128
	2nd	2	[128, 128]	256
10	1st	2	[32, 32]	64
	2nd	3	[64, 64, 64]	64
16	1st	2	[32, 32]	128
	2nd	3	[32, 32, 32]	64
39	1st	2	[32, 32]	64
	2nd	3	[64, 64, 64]	256
47	1st	3	[64, 64, 64]	32
	2nd	2	[32, 32]	128
53	1st	2	[32, 32]	128
	2nd	2	[128, 128]	256
67	1st	2	[32, 32]	256
	2nd	3	[64, 64, 64]	32
59	1st	2	[128, 128]	64
	2nd	2	[32, 32]	64
74	1st	3	[256, 256, 256]	64
	2nd	2	[128, 128]	256
81	1st	2	[32, 32]	128
	2nd	2	[128, 128]	64

Appendix E

Day-time-only forecast results

E.1 Forecast results

With the day-time-only results obtained for the inverter-level and macro-level PV-system forecast models, Figure E.1 demonstrates the CI distributions obtained for each day type. Once again, forecast hours 1 h, 3 h and 6 h ahead are displayed due to the evident transition in forecasting accuracy over these forecast time periods.

For a more detailed overview of the obtained CI metrics, this information is summarised for each day type in Table E.1 - Table E.4.

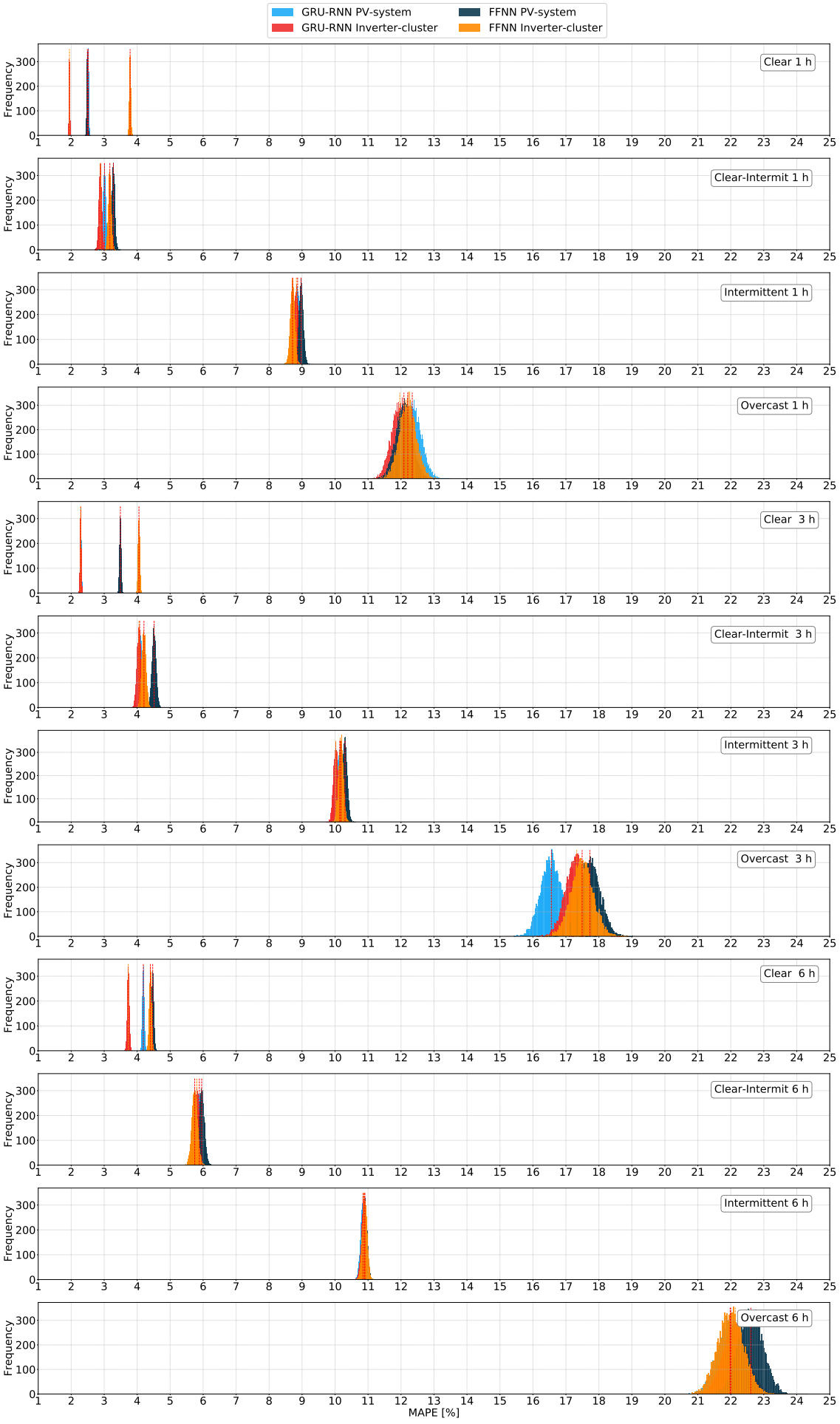


Figure E.1: Day-time-only test set MAPE Bootstrap CIs for the GRU-RNN and FFNN aggregated inverter-level and macro-level forecast accuracies.

Table E.1: Day-time-only 95 % Bootstrap CIs obtained for clear weather conditions for the aggregated inverter-level and macro-level PV-system FFNN and GRU-RNN models.

Forecast model		Avg. MAPE [%]	Lower Bound [%]	Upper Bound [%]	CI width [kW]
			1 h		
GRU-RNN	PV-system	2.52	2.49	2.55	42.84
	Inverter cluster	1.95	1.92	1.97	34.99
FFNN	PV-system	2.49	2.46	2.52	46.60
	Inverter cluster	3.79	3.75	3.83	61.47
			3 h		
GRU-RNN	PV-system	2.29	2.25	2.33	55.35
	Inverter cluster	2.29	2.25	2.32	51.27
FFNN	PV-system	3.49	3.44	3.54	70.55
	Inverter cluster	4.06	4.01	4.10	73.41
			6 h		
GRU-RNN	PV-system	4.19	4.13	4.24	80.84
	Inverter cluster	3.73	3.68	3.79	84.88
FFNN	PV-system	4.47	4.40	4.54	101.71
	Inverter cluster	4.40	4.34	4.47	98.30

Table E.2: Day-time-only 95 % Bootstrap CIs obtained for clear-intermittent weather conditions for the aggregated inverter-level and macro-level PV-system FFNN and GRU-RNN models.

Forecast model		Avg. MAPE [%]	Lower Bound [%]	Upper Bound [%]	CI width [kW]
			1 h		
GRU-RNN	PV-system	3.01	2.93	3.10	127.24
	Inverter cluster	2.88	2.80	2.97	121.63
FFNN	PV-system	3.28	3.20	3.36	126.04
	Inverter cluster	3.17	3.09	3.26	124.70
			3 h		
GRU-RNN	PV-system	4.08	3.95	4.20	186.46
	Inverter cluster	4.06	3.94	4.18	182.89
FFNN	PV-system	4.51	4.39	4.64	185.34
	Inverter cluster	4.21	4.08	4.33	184.84
			6 h		
GRU-RNN	PV-system	5.88	5.73	6.04	231.83
	Inverter cluster	5.81	5.65	5.98	245.43
FFNN	PV-system	5.96	5.80	6.12	244.69
	Inverter cluster	5.75	5.58	5.91	251.05

Table E.3: Day-time-only 95 % Bootstrap CIs obtained for intermittent weather conditions for the aggregated inverter-level and macro-level PV-system FFNN and GRU-RNN models.

Forecast model		Avg. MAPE [%]	Lower Bound [%]	Upper Bound [%]	CI width [kW]
			1 h		
GRU-RNN	PV-system	8.86	8.74	9.00	194.23
	Inverter cluster	8.83	8.70	8.95	188.23
FFNN	PV-system	8.97	8.84	9.10	189.69
	Inverter cluster	8.71	8.58	8.84	192.39
			3 h		
GRU-RNN	PV-system	10.14	10.01	10.28	204.65
	Inverter cluster	10.02	9.89	10.16	202.86
FFNN	PV-system	10.30	10.16	10.44	206.61
	Inverter cluster	10.18	10.04	10.32	207.39
			6 h		
GRU-RNN	PV-system	10.85	10.70	10.99	216.74
	Inverter cluster	10.87	10.73	11.01	211.77
FFNN	PV-system	10.90	10.76	11.04	212.87
	Inverter cluster	10.90	10.76	11.04	212.21

Table E.4: Day-time-only 95 % Bootstrap CIs obtained for overcast weather conditions for the aggregated inverter-level and macro-level PV-system FFNN and GRU-RNN models.

Forecast model		Avg. MAPE [%]	Lower Bound [%]	Upper Bound [%]	CI width [kW]
			1 h		
GRU-RNN	PV-system	12.34	11.82	12.88	796.94
	Inverter cluster	11.96	11.46	12.48	767.36
FFNN	PV-system	12.09	11.60	12.59	745.07
	Inverter cluster	12.21	11.70	12.72	763.04
			3 h		
GRU-RNN	PV-system	16.56	15.96	17.18	912.55
	Inverter cluster	17.33	16.70	17.97	951.49
FFNN	PV-system	17.73	17.10	18.35	935.13
	Inverter cluster	17.49	16.84	18.14	976.11
			6 h		
GRU-RNN	PV-system	21.98	21.23	22.75	1138.51
	Inverter cluster	22.14	21.43	22.87	1081.75
FFNN	PV-system	22.61	21.88	23.33	1091.13
	Inverter cluster	22.00	21.24	22.75	1134.48

Appendix F

Training time

F.1 Forecast model training time

The total macro-level and aggregated inverter-level model training times are presented in Table F.1. Evidently, the aggregated inverter-level forecasts have a much higher computational expense regarding the time requirement. This can be expected, since 84 inverter models had to be trained in comparison to only one macro-level model.

Table F.1: Total training time required for macro-level and inverter-level models.

	Total training time [min]	
Forecast model	**FFNN**	**GRU-RNN**
Macro-level PV-system model	5.06	11.48
Aggregated inverter-level model	505.96	3483.88

It should be mentioned that these training times are defined for a scenario where only one NVIDIA Titan Xp GPU card is available for training. Fortunately, the actual training time for the execution of the inverter-level models could be reduced by a factor of two, due to the availability of two GPUs used to train the models in parallel.

Appendix G

Inverter clustering

G.1 Defining the number of clusters

With the K-Means clustering algorithm, it is necessary to define the number of clusters. However, it is not always a completely obvious decision, since the clustering applied for this research is an unsupervised training process. With the use of the Elbow method [134], the CH-index [135] and the Gap statistic [136] as guidelines, some indications of the number of clusters to be applied were obtained. In addition to this, the use of the Silhouette, Dunn and Davies-Bouldin index cluster evaluation metrics were also applied. These three metrics indicate how appropriate the selection of clusters is (also referred to as cluster validity).

From the Elbow method presented in Figure G.1, the red-line slope intersects at 10 clusters. From the the CH-index in Figure G.2, K is also identified in the region of approximately 10 clusters. Finally, with the use of the Gap statistic [136] in Figure G.3, there is also some indication that 10 clusters should be adequate.

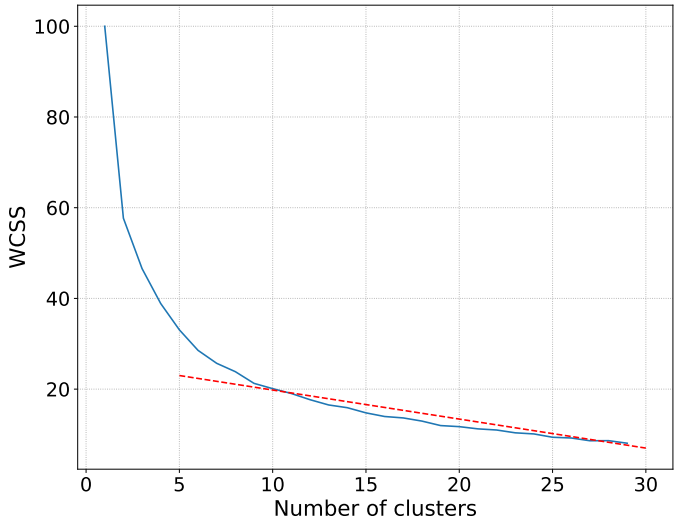


Figure G.1: Number of inverter-clusters to be allocated as indicated Elbow method.

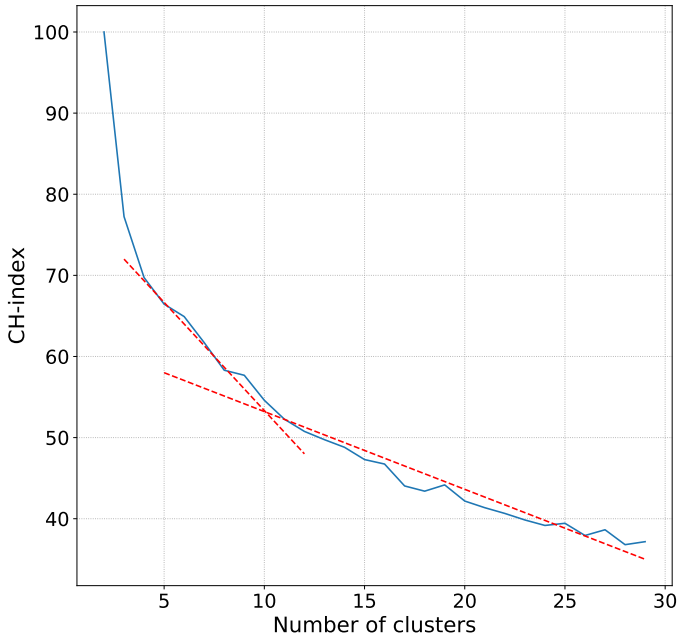


Figure G.2: Number of inverter-clusters to be allocated as indicated CH-index.

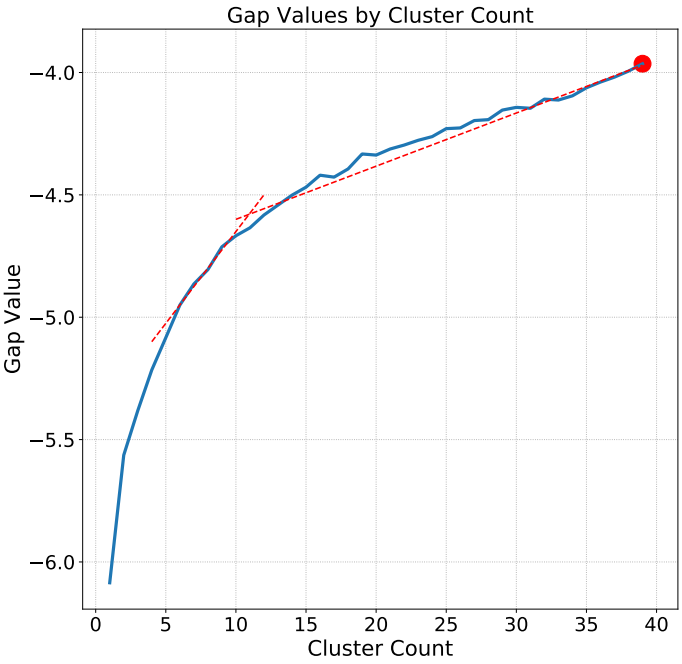


Figure G.3: Number of inverter-clusters to be allocated as indicated with Gap statistic.

The application of the well documented Dunn index [142], which serves as metric for evaluating the number of clusters, is presented in Figure G.4. A high Dunn index value is indicative of a good cluster allocation. This metric rewards a compact distribution of samples within the clusters, where the means of each cluster are far apart (well separated clusters) [143, p.389]. From Figure G.4, the Dunn index values increase as the number of clusters increase. This is similar to the behaviour seen as for example with the CH-index and Elbow methods. However, for the purpose of optimisation, a minimum number of clusters must be selected. Evidently, there is a jump in performance from a selection of six clusters to 10 clusters, with cluster values ranging between 10 - 17 delivering a relatively stable Dunn index measure.

A further improvement is seen for the selection of clusters ranging between about 20 - 26 clusters. However, the sporadic changes in the Dunn index values for the clusters ranging between 18 - 30 provide somewhat uncertainty regarding an exact choice.

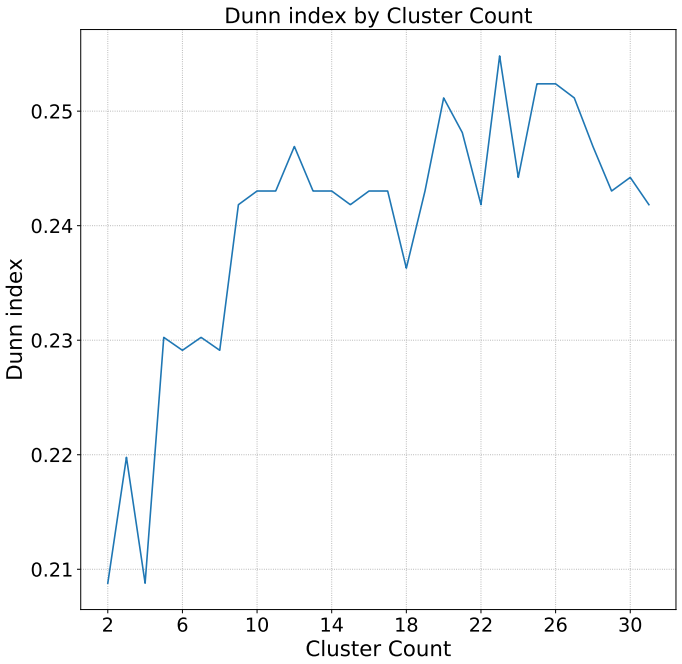


Figure G.4: Dunn index cluster evaluation metric results.

The Davies-Bouldin index [144] is another well referenced cluster evaluation metric. This metric measures the ratio of the sum of within-cluster scatter to between cluster separation [145]. Therefore, a lower Davies-Bouldin measure is indicative of a better selection of clusters. Figure G.5 illustrates the Davies-Bouldin measures, obtained for each cluster. From this graph, similar to the Dunn index values, there is a definitive improvement from six clusters towards 10 clusters. However, once again the clustering results improve as more clusters are applied, with 25 - 30 clusters delivering a range with the lowest Davies-Bouldin values.

It is important to understand that these are merely guidelines, which serve to identify a viable range of possible clusters. There is no exact theory to truly support the use of these techniques in a generic sense. From further analysis, given the results of all of the above clustering evaluations, there is a definite inflection point for an allocation of 10 clusters. Reason being that a selection of 10 clusters has delivered a noteworthy performance increase in all instances, as compared to a smaller selection of clusters.

However, considering the Dunn and Davies-Bouldin indexes it must be mentioned that a selection of 20 clusters is also indicative of potentially delivering good results. However, given the Elbow and CH-index results, together with the objective of optimisation, the lowest number of clusters had to be selected.

Therefore, the selection of K as 10 clusters was further investigated as a viable solution. From this empirical analysis, which was presented in Section 4.4.2, it was shown that these guidelines did in fact prove to be valuable, with the inverter clustering further demonstrated to be sufficient for 10 clusters. As further concluded from the results, the selection of 10 clusters, in comparison to a scenario where all 84 inverters were individually optimised for both hyperparameter and parameter selections, delivered an almost exact forecasting accuracy. Therefore, proving the selection of 10 clusters as valid for the intent of this research.

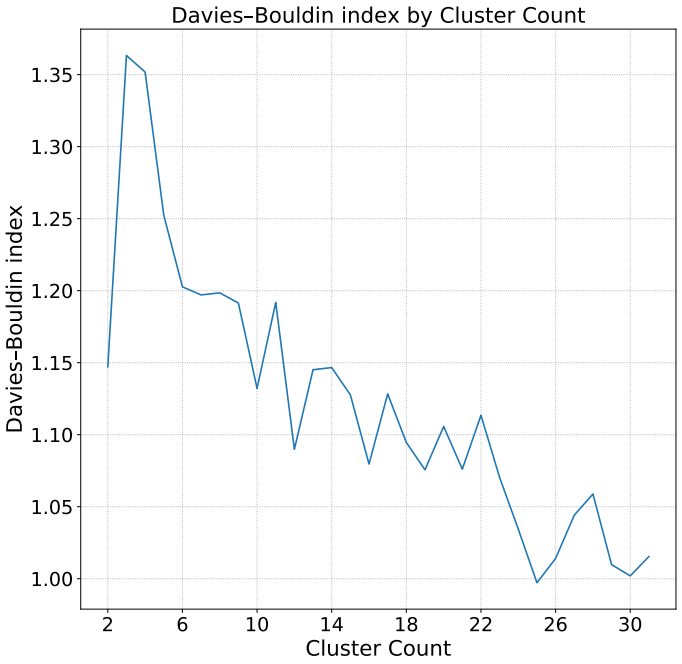


Figure G.5: Davies-Bouldin index cluster evaluation metric results.

List of References

[1] REN21, "Renewables 2020 global status report," Paris, France, 2020. [Online]. Available: https://www.ren21.net/gsr-2020/

[2] F. Barbieri, S. Rajakaruna, and A. Ghosh, "Very short-term photovoltaic power forecasting with cloud modeling: A review," *Renewable and Sustainable Energy Reviews*, vol. 75, pp. 242 – 263, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S136403211630733X

[3] D. Masa-Bote, M. Castillo-Cagigal, E. Matallanas, E. C. no Martín, A. Gutiérrez, F. Monasterio-Huelín, and J. Jiménez-Leube, "Improving photovoltaics grid integration through short time forecasting and self-consumption," *Applied Energy*, vol. 125, no. Supplement C, pp. 103 – 113, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261914002761

[4] P. Mandal, S. T. S. Madhira, A. U. haque, J. Meng, and R. L. Pineda, "Forecasting power output of solar photovoltaic system using wavelet transform and artificial intelligence techniques," *Procedia Computer Science*, vol. 12, pp. 332–337, 2012. [Online]. Available: https://doi.org/10.1016/j.procs.2012.09.080

[5] Z. Zheng, Y. Chen, M. Huo, and B. Zhao, "An overview: the development of prediction technology of wind and photovoltaic power generation," *Energy Procedia*, vol. 12, pp. 601–608, 2011. [Online]. Available: https://doi.org/10.1016/j.egypro.2011.10.081

[6] L. A. Fernandez-Jimenez, A. M. noz Jimenez, A. Falces, M. Mendoza-Villena, E. Garcia-Garrido, P. M. Lara-Santillan, E. Zorzano-Alba, and P. J. Zorzano-Santamaria, "Short-term power forecasting system for photovoltaic plants," *Renewable Energy*, vol. 44, pp. 311 – 317, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960148112001516

[7] "Eex homepage," http://www.eex.com/en/, (Accessed on 12/16/2017).

[8] A. A. S. de la Nieta, N. G. Paterakis, and M. Gibescu, "Participation of photovoltaic power producers in short-term electricity markets based on rescheduling and risk-hedging mapping," *Applied Energy*, vol. 266, p. 114741, 2020.

[9] Y.-K. Wu, C.-R. Chen, and H. A. Rahman, "A novel hybrid model for short-term forecasting in PV power generation," *International Journal of Photoenergy*, vol. 2014, pp. 1–9, 2014. [Online]. Available: https://doi.org/10.1155/2014/569249

[10] S. K. Chow, E. W. Lee, and D. H. Li, "Short-term prediction of photovoltaic energy generation by intelligent approach," *Energy and Buildings*, vol. 55, pp. 660–667, dec 2012. [Online]. Available: https://doi.org/10.1016/j.enbuild.2012.08.011

[11] A. Mellit, A. M. Pavan, and V. Lughi, "Short-term forecasting of power production in a large-scale photovoltaic plant," *Solar Energy*, vol. 105, pp. 401 – 413, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X14001522

[12] A. Mellit and S. A. Kalogirou, "Artificial intelligence techniques for photovoltaic applications: A review," *Progress in Energy and Combustion Science*, vol. 34, no. 5, pp. 574 – 632, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360128508000026

[13] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. M. de Pison, and F. Antonanzas-Torres, "Review of photovoltaic power forecasting," *Solar Energy*, vol. 136, pp. 78 – 111, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X1630250X

[14] M. Q. Raza, M. Nadarajah, and C. Ekanayake, "On recent advances in PV output power forecast," *Solar Energy*, vol. 136, pp. 125–144, oct 2016. [Online]. Available: https://doi.org/10.1016/j.solener.2016.06.073

[15] U. K. Das, K. S. Tey, M. Seyedmahmoudian, S. Mekhilef, M. Y. I. Idris, W. V. Deventer, B. Horan, and A. Stojcevski, "Forecasting of photovoltaic power generation and model optimization: A review," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 912–928, jan 2018. [Online]. Available: https://doi.org/10.1016/j.rser.2017.08.017

[16] S. Sobri, S. Koohi-Kamali, and N. A. Rahim, "Solar photovoltaic generation forecasting methods: A review," *Energy Conversion and Management*, vol. 156, pp. 459–497, 2018.

[17] D. van der Meer, J. Widén, and J. Munkhammar, "Review on probabilistic forecasting of photovoltaic power production and electricity consumption," *Renewable and Sustainable Energy Reviews*, vol. 81, no. Part 1, pp. 1484 – 1512, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032117308523

[18] A. Mellit, A. Massi Pavan, E. Ogliari, S. Leva, and V. Lughi, "Advanced methods for photovoltaic output power forecasting: A review," p. 487, 2020.

[19] R. Ahmed, V. Sreeram, Y. Mishra, and M. Arif, "A review and evaluation of the state-of-the-art in pv solar power forecasting: Techniques and optimization," *Renewable and Sustainable Energy Reviews*, vol. 124, p. 109792, 2020.

[20] V. Kostylev, A. Pavlovski *et al.*, "Solar power forecasting performance-towards industry standards," in *1st International Workshop on the Integration of Solar Power into Power Systems, Aarhus, Denmark*, 2011.

[21] S. Daliento, A. Chouder, P. Guerriero, A. M. Pavan, A. Mellit, R. Moeini, and P. Tricoli, "Monitoring, diagnosis, and power forecasting for photovoltaic fields: A review," *International Journal of Photoenergy*, vol. 2017, pp. 1–13, 2017. [Online]. Available: https://doi.org/10.1155/2017/1356851

[22] J. Kleissl, *Solar Energy Forecasting and Resource Assessment*. Academic Press, 2013. [Online]. Available: https://www.amazon.com/Solar-Energy-Forecasting-Resource-Assessment-ebook/dp/B00DRF0HZS?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=B00DRF0HZS

[23] C. Wan, J. Zhao, Y. Song, Z. Xu, J. Lin, and Z. Hu, "Photovoltaic and solar power forecasting for smart grid energy management," *CSEE Journal of Power and Energy Systems*, vol. 1, no. 4, pp. 38–46, Dec 2015.

[24] A. Nespoli, E. Ogliari, S. Leva, A. Massi Pavan, A. Mellit, V. Lughi, and A. Dolara, "Day-ahead photovoltaic forecasting: A comparison of the most effective techniques," *Energies*, vol. 12, no. 9, p. 1621, 2019.

[25] C. Voyant, G. Notton, S. Kalogirou, M.-L. Nivet, C. Paoli, F. Motte, and A. Fouilloy, "Machine learning methods for solar radiation forecasting: A review," *Renewable Energy*, vol. 105, pp. 569 – 582, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960148116311648

[26] R. H. Inman, H. T. Pedro, and C. F. Coimbra, "Solar forecasting methods for renewable energy integration," *Progress in Energy and Combustion Science*, vol. 39, no. 6, pp. 535 – 576, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360128513000294

[27] D. Anagnostos, T. Schmidt, S. Cavadias, D. Soudris, J. Poortmans, and F. Catthoor, "A method for detailed, short-term energy yield forecasting of photovoltaic installations," *Renewable Energy*, vol. 130, pp. 122–129, 2019.

[28] M. Rana, I. Koprinska, and V. G. Agelidis, "Univariate and multivariate methods for very short-term solar photovoltaic power forecasting," *Energy Conversion and Management*, vol. 121, no. Supplement C, pp. 380 – 390, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0196890416303934

[29] M. Abdel-Nasser and K. Mahmoud, "Accurate photovoltaic power forecasting models using deep lstm-rnn," *Neural Computing and Applications*, vol. 31, no. 7, pp. 2727–2740, 2019.

[30] Y. Chu, B. Urquhart, S. M. Gohari, H. T. Pedro, J. Kleissl, and C. F. Coimbra, "Short-term reforecasting of power output from a 48 mw solar pv plant," *Solar Energy*, vol. 112, pp. 68 – 77, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X14005611

[31] M. Diagne, M. David, P. Lauret, J. Boland, and N. Schmutz, "Review of solar irradiance forecasting methods and a proposition for small-scale insular grids," *Renewable and Sustainable Energy Reviews*, vol. 27, pp. 65 – 76, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032113004334

[32] J. Zhang, A. Florita, B.-M. Hodge, S. Lu, H. F. Hamann, V. Banunarayanan, and A. M. Brockway, "A suite of metrics for assessing the performance of solar power forecasting," *Solar Energy*, vol. 111, pp. 157 – 175, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X14005027

[33] X. Zhao, H. A. Antwi, and E. Yiranbon, "Forecasting optimal solar energy supply in jiangsu province (china): A systematic approach using hybrid of weather and energy forecast models," *The Scientific World Journal*, vol. 2014, pp. 1–9, 2014. [Online]. Available: https://doi.org/10.1155/2014/580606

[34] T. E. Hoff, R. Perez, J. Kleissl, D. Renne, and J. Stein, "Reporting of irradiance modeling relative prediction errors," *Progress in Photovoltaics: Research and Applications*, vol. 21, no. 7, pp. 1514–1519, 2013.

[35] H. Wang, H. Yi, J. Peng, G. Wang, Y. Liu, H. Jiang, and W. Liu, "Deterministic and probabilistic forecasting of photovoltaic power based on deep convolutional neural network," *Energy conversion and management*, vol. 153, pp. 409–422, 2017.

[36] Q. Ni, S. Zhuang, H. Sheng, G. Kang, and J. Xiao, "An ensemble prediction intervals approach for short-term pv power forecasting," *Solar Energy*, vol. 155, pp. 1072 – 1083, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X17306370

[37] H. T. Pedro, C. F. Coimbra, M. David, and P. Lauret, "Assessment of machine learning techniques for deterministic and probabilistic intra-hour solar forecasts," *Renewable Energy*, vol. 123, pp. 191–203, 2018.

[38] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE transactions on neural networks*, vol. 22, no. 3, pp. 337–346, 2010.

[39] H. Quan, D. Srinivasan, and A. Khosravi, "Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 303–315, Feb 2014.

[40] S. Sperati, S. Alessandrini, and L. Delle Monache, "An application of the ecmwf ensemble prediction system for short-term solar power forecasting," *Solar Energy*, vol. 133, pp. 437–450, 2016.

[41] Q. Ni, S. Zhuang, H. Sheng, G. Kang, and J. Xiao, "An ensemble prediction intervals approach for short-term pv power forecasting," *Solar Energy*, vol. 155, pp. 1072–1083, 2017.

[42] A. Khosravi and S. Nahavandi, "Combined nonparametric prediction intervals for wind power generation," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 4, pp. 849–856, 2013.

[43] E. Camporeale, X. Chu, O. Agapitov, and J. Bortnik, "On the generation of probabilistic forecasts from deterministic models," *Space Weather*, vol. 17, no. 3, pp. 455–475, 2019.

[44] K. Wang, X. Qi, and H. Liu, "A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network," *Applied Energy*, vol. 251, p. 113315, 2019.

[45] H.-z. Wang, G.-q. Li, G.-b. Wang, J.-c. Peng, H. Jiang, and Y.-t. Liu, "Deep learning based ensemble approach for probabilistic wind power forecasting," *Applied energy*, vol. 188, pp. 56–70, 2017.

[46] A. Khosravi, S. Nahavandi, and D. Creighton, "Prediction intervals for short-term wind farm power generation forecasts," *IEEE Transactions on sustainable energy*, vol. 4, no. 3, pp. 602–610, 2013.

[47] G. Nason, *Wavelet Methods in Statistics with R (Use R!)*. Springer, 2013. [Online]. Available: https://www.amazon.com/Wavelet-Methods-Statistics-Use-Nason-ebook/dp/B00DZ1174E?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=B00DZ1174E

[48] H. T. Pedro and C. F. Coimbra, "Assessment of forecasting techniques for solar power production with no exogenous inputs," *Solar Energy*, vol. 86, no. 7, pp. 2017 – 2028, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X12001429

[49] H. Zhou, Y. Zhang, L. Yang, Q. Liu, K. Yan, and Y. Du, "Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism," *IEEE Access*, vol. 7, pp. 78 063–78 074, 2019.

[50] H. Sharadga, S. Hajimirza, and R. S. Balog, "Time series forecasting of solar power generation for large-scale photovoltaic plants," *Renewable Energy*, vol. 150, pp. 797–807, 2020.

[51] H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng, "A review of deep learning for renewable energy forecasting," *Energy Conversion and Management*, vol. 198, p. 111799, 2019.

[52] K. Li, R. Wang, H. Lei, T. Zhang, Y. Liu, and X. Zheng, "Interval prediction of solar power using an improved bootstrap method," *Solar Energy*, vol. 159, pp. 97 – 112, 2018.

[53] M. Hossain, S. Mekhilef, M. Danesh, L. Olatomiwa, and S. Shamshirband, "Application of extreme learning machine for short term output power forecasting of three grid-connected pv systems," *Journal of Cleaner Production*, vol. 167, pp. 395 – 405, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959652617317973

[54] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[55] M. G. D. Giorgi, P. M. Congedo, M. Malvoni, and D. Laforgia, "Error analysis of hybrid photovoltaic power forecasting models: A case study of mediterranean climate," *Energy Conversion and Management*, vol. 100, pp. 117 – 130, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0196890415004422

[56] D. AlHakeem, P. Mandal, A. U. Haque, A. Yona, T. Senjyu, and T. L. Tseng, "A new strategy to quantify uncertainties of wavelet-grnn-pso based solar pv power forecasts using bootstrap confidence intervals," in *2015 IEEE Power Energy Society General Meeting*, July 2015, pp. 1–5.

[57] P. H. Chiang, S. P. V. Chiluvuri, S. Dey, and T. Q. Nguyen, "Forecasting of solar photovoltaic system power generation using wavelet decomposition and bias-compensated random forest," in *2017 Ninth Annual IEEE Green Technologies Conference (GreenTech)*, March 2017, pp. 260–266.

[58] C. Persson, P. Bacher, T. Shiga, and H. Madsen, "Multi-site solar power forecasting using gradient boosted regression trees," *Solar Energy*, vol. 150, pp. 423 – 436, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X17303717

[59] X. G. Agoua, R. Girard, and G. Kariniotakis, "Short-term spatio-temporal forecasting of photovoltaic power production," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 2, pp. 538–546, April 2018.

[60] L. Liu, Y. Zhao, D. Chang, J. Xie, Z. Ma, Q. Sun, H. Yin, and R. Wennersten, "Prediction of short-term pv power output and uncertainty analysis," *Applied Energy*, vol. 228, pp. 700 – 711, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261918309826

[61] Z. Li, S. Rahman, R. Vega, and B. Dong, "A hierarchical approach using machine learning methods in solar photovoltaic energy production forecasting," *Energies*, vol. 9, no. 1, p. 55, jan 2016. [Online]. Available: https://doi.org/10.3390/en9010055

[62] J. Li, J. K. Ward, J. Tong, L. Collins, and G. Platt, "Machine learning for solar irradiance forecasting of photovoltaic system," *Renewable Energy*, vol. 90, pp. 542 – 553, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960148115305747

[63] H. Zhu, X. Li, Q. Sun, L. Nie, J. Yao, and G. Zhao, "A power prediction method for photovoltaic power plant based on wavelet decomposition and artificial neural networks," *Energies*, vol. 9, no. 1, p. 11, dec 2015. [Online]. Available: https://doi.org/10.3390/en9010011

[64] M. Bouzerdoum, A. Mellit, and A. M. Pavan, "A hybrid model (sarima-svm) for short-term power forecasting of a small-scale grid-connected photovoltaic plant," *Solar Energy*, vol. 98, pp. 226 – 235, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X13004039

[65] W. Lee, K. Kim, J. Park, J. Kim, and Y. Kim, "Forecasting solar power using long-short term memory and convolutional neural networks," *IEEE Access*, vol. 6, pp. 73 068–73 080, November 2018.

[66] C. Monteiro, T. Santos, L. Fernandez-Jimenez, I. Ramirez-Rosado, and M. Terreros-Olarte, "Short-term power forecasting model for photovoltaic plants based on historical similarity," *Energies*, vol. 6, no. 5, pp. 2624–2643, may 2013. [Online]. Available: https://doi.org/10.3390/en6052624

[67] M. Rana, I. Koprinska, and V. G. Agelidis, "2d-interval forecasts for solar power production," *Solar Energy*, vol. 122, no. Supplement C, pp. 191 – 203, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X15004545

[68] M. Malvoni, M. G. De Giorgi, and P. M. Congedo, "Forecasting of pv power generation using weather input data-preprocessing techniques," *Energy Procedia*, vol. 126, pp. 651–658, 2017.

[69] A. T. Eseye, J. Zhang, and D. Zheng, "Short-term photovoltaic solar power forecasting using a hybrid wavelet-pso-svm model based on scada and meteorological information," *Renewable Energy*, vol. 118, pp. 357–367, 2018.

[70] F. Chollet, *Deep learning with Python*, 2018.

[71] D. Lee and K. Kim, "Recurrent neural network-based hourly prediction of photovoltaic power output using meteorological information," *Energies*, vol. 12, no. 2, p. 215, 2019.

[72] P. Li, K. Zhou, X. Lu, and S. Yang, "A hybrid deep learning model for short-term pv power forecasting," *Applied Energy*, vol. 259, p. 114216, 2020.

[73] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[74] Y. Wang, W. Liao, and Y. Chang, "Gated recurrent unit network-based short-term photovoltaic forecasting," *Energies*, vol. 11, no. 8, p. 2163, 2018.

[75] N. Sodsong, K. M. Yu, and W. Ouyang, "Short-term solar pv forecasting using gated recurrent unit with a cascade model," in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Feb 2019, pp. 292–297.

[76] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting-an approach using autoencoder and lstm neural networks," in *2016 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2016, pp. 002 858–002 865.

[77] B. Urquhart, M. Ghonima, D. A. Nguyen, B. Kurtz, C. W. Chow, and J. Kleissl, "Chapter 9 - sky-imaging systems for short-term forecasting," in *Solar Energy Forecasting and Resource Assessment*, J. Kleissl, Ed. Boston: Academic Press, 2013, pp. 195 – 232. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123971777000097

[78] M. Lipperheide, J. Bosch, and J. Kleissl, "Embedded nowcasting method using cloud speed persistence for a photovoltaic power plant," *Solar Energy*, vol. 112, pp. 232 – 238, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0038092X1400557X

[79] J. Liu, W. Fang, X. Zhang, and C. Yang, "An improved photovoltaic power forecasting model with the assistance of aerosol index data," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 2, pp. 434–442, April 2015.

[80] L. Gigoni, A. Betti, E. Crisostomi, A. Franco, M. Tucci, F. Bizzarri, and D. Mucci, "Day-ahead hourly forecasting of power generation from photovoltaic plants," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 2, pp. 831–842, 2017.

[81] M. Pierro, M. De Felice, E. Maggioni, D. Moser, A. Perotto, F. Spada, and C. Cornaro, "Data-driven upscaling methods for regional photovoltaic power estimation and forecast using satellite and numerical weather prediction data," *Solar Energy*, vol. 158, pp. 1026–1038, 2017.

[82] G. M. Yagli, D. Yang, and D. Srinivasan, "Automatic hourly solar forecasting using machine learning models," *Renewable and Sustainable Energy Reviews*, vol. 105, pp. 487–498, 2019.

[83] M. Husein and I.-Y. Chung, "Day-ahead solar irradiance forecasting for microgrids using a long short-term memory recurrent neural network: A deep learning approach," *Energies*, vol. 12, no. 10, p. 1856, 2019.

[84] S. Sala, A. Amendola, S. Leva, M. Mussetta, A. Niccolai, and E. Ogliari, "Comparison of data-driven techniques for nowcasting applied to an industrial-scale photovoltaic plant," *Energies*, vol. 12, no. 23, p. 4520, 2019.

[85] G. Makrides, B. Zinsser, A. Phinikarides, M. Schubert, and G. E. Georghiou, "Temperature and thermal annealing effects on different photovoltaic technologies," *Renewable Energy*, vol. 43, pp. 407–417, 2012.

[86] W. Charles Lawrence Kamuyu, J. R. Lim, C. S. Won, and H. K. Ahn, "Prediction model of photovoltaic module temperature for power performance of floating pvs," *Energies*, vol. 11, no. 2, p. 447, 2018.

[87] C. Schwingshackl, M. Petitta, J. Wagner, G. Belluardo, D. Moser, M. Castelli, M. Zebisch, and A. Tetzlaff, "Wind effect on pv module temperature: Analysis of different techniques for an accurate estimation," *Energy Procedia*, vol. 40, no. Supplement C, pp. 77 – 86, 2013, european Geosciences Union General Assembly 2013, EGUDivision Energy, Resources & the Environment, ERE. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1876610213016044

[88] J. K. Kaldellis, M. Kapsali, and K. A. Kavadias, "Temperature and wind speed impact on the efficiency of pv installations. experience obtained from outdoor measurements in greece," *Renewable Energy*, vol. 66, no. Supplement C, pp. 612 – 624, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960148114000123

[89] S. Solar, "South africa - scatec solar," https://scatecsolar.com/locations/south-africa/#south-africa, (Accessed on 06/22/2020).

[90] "Les mées solar farm - france," http://projects.mcrit.com/esponfutures/index.php/principal/53-the-les-mees-solar-farm-france, 2011, (Accessed on 10/10/2020).

[91] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015.

[92] Y. Ren, P. Suganthan, and N. Srikanth, "Ensemble methods for wind and solar power forecasting-a state-of-the-art review," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 82–91, 2015.

[93] "List of power stations in south africa - wikipedia," https://en.wikipedia.org/wiki/List_of_power_stations_in_South_Africa#Solar_PV_power, (Accessed on 07/26/2020).

[94] S. V. Szokolay, "Solar Geometry," *Design Tools and Techniques*, vol. 53, p. 160, 1989.

[95] G. M. Masters, *Renewable And Efficient Electric Power Systems*, 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2013.

[96] R. Andrews, W. Holmgren, T. Lorenzo, and C. Hansen, "get_solarposition function: pvlib-python/solarposition.py at master pvlib/pvlib-python github," https://github.com/pvlib/pvlib-python/blob/master/pvlib/solarposition.py, 2018, (Accessed on 09/26/2020).

[97] C. Cornwall, A. Horiuchi, and C. Lehman, "NOAA Earth System Research Laboratory. Sun Position Calculator." [Online]. Available: http://nomads.ncep.noaa.gov/

[98] A. Zheng and A. Casari, *Feature engineering for machine learning: principles and techniques for data scientists.* " O'Reilly Media, Inc.", 2018.

[99] J. Lawrence, *Introduction To Neural Networks: Design, Theory, and Applications, Sixth Edition*. California Scientific Software, 1994. [Online]. Available: https://www.amazon.com/Introduction-Neural-Networks-Design-Applications/dp/1883157005?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1883157005

[100] C. Tosh and G. Ruxton, *Modelling perception with artificial neural networks*. Cambridge, UK ; New York: Cambridge University Press, 2010.

[101] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[102] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[103] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[104] X. Jin, C. Xu, J. Feng, Y. Wei, J. Xiong, and S. Yan, "Deep learning with s-shaped rectified linear activation units," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[105] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Advances in neural information processing systems*, 2017, pp. 971–980.

[106] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.

[107] J. R. Rabunal, *Artificial Neural Networks in Real-Life Applications*. IGI Global, 2005. [Online]. Available: https://www.amazon.com/Artificial-Neural-Networks-Real-Life-Applications/dp/1591409020?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1591409020

[108] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[109] H. Kamper, "Yet another introduction to backpropagation," https://www.kamperh.com/notes/kamper_backprop17.pdf, November 2017, (Accessed on 02/13/2020).

[110] C. Olah, "Understanding lstm networks, 2015," *URL http://colah. github. io/posts/2015-08-Understanding-LSTMs*, 2015, (Accessed on 02/13/2020).

[111] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[112] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *International conference on machine learning*, 2015, pp. 2342–2350.

[113] J. Brownlee, *Long Short-term Memory Networks with Python: Develop Sequence Prediction Models with Deep Learning*. Machine Learning Mastery, 2017.

[114] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2017.

[115] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[116] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[117] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[118] A. Graves, M. Liwicki, S. FernÃ¡ndez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, May 2009.

[119] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[120] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[121] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[122] P. R. Hinton, I. McMurray, and C. Brownlow, *SPSS explained*. Routledge, 2014.

[123] S.-D. Bolboaca and L. Jäntschi, "Pearson versus spearman, kendall's tau correlation analysis on structure-activity relationships of biologic active compounds," *Leonardo Journal of Sciences*, vol. 5, no. 9, pp. 179–200, 2006.

[124] J. Hauke and T. Kossowski, "Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data," *Quaestiones geographicae*, vol. 30, no. 2, pp. 87–93, 2011.

[125] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[126] D. Masters and C. Luschi, "Revisiting small batch training for deep neural networks," *CoRR*, vol. abs/1804.07612, 2018. [Online]. Available: http://arxiv.org/abs/1804.07612

[127] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of machine learning research*, vol. 13, no. Feb, pp. 281–305, 2012.

[128] D. R. Wilson and T. R. Martinez, "The general inefficiency of batch training for gradient descent learning," *Neural networks*, vol. 16, no. 10, pp. 1429–1451, 2003.

[129] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: a survey and empirical demonstration," *Data Mining and knowledge discovery*, vol. 7, no. 4, pp. 349–371, 2003.

[130] P. Senin, "Dynamic time warping algorithm review," *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, vol. 855, no. 1-23, p. 40, 2008.

[131] O. Lauwers and B. De Moor, "A time series distance measure for efficient clustering of input/output signals by their underlying dynamics," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 286–291, 2017.

[132] E. Keogh and J. Lin, "Clustering of time-series subsequences is meaningless: implications for previous and future research," *Knowledge and information systems*, vol. 8, no. 2, pp. 154–177, 2005.

[133] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*, ser. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011, ch. 10.2.

[134] R. Bonnin, *Machine Learning for Developers: Uplift your regular applications with the power of statistics, analytics, and machine learning*. Packt Publishing, 2017, pp. 78,79.

[135] A. Nasser, "Investigating k-means and kernel k-means algorithms with internal validity indices for cluster identification," *Journal of Advances in Mathematics and Computer Science*, pp. 1–12, 2019.

[136] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[137] J. G. Carney, P. Cunningham, and U. Bhagwan, "Confidence and prediction intervals for neural network ensembles," in *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, vol. 2. IEEE, 1999, pp. 1215–1218.

[138] J. S. Haukoos and R. J. Lewis, "Advanced statistics: bootstrapping confidence intervals for statistics with difficult distributions," *Academic emergency medicine*, vol. 12, no. 4, pp. 360–365, 2005.

[139] C. Plus, "Taihang mountain goes solar - china plus," http://chinaplus.cri.cn/photo/china/18/20181218/224696_1.html, December 2018, (Accessed on 10/22/2020).

[140] R. A. Messenger and J. Ventre, *Photovoltaic Systems Engineering*, 3rd ed. New York, New York: CRC Press Francis and Taylor Group, 2010.

[141] I. Reda and A. Andreas, "Solar Position Algorithm for Solar Radiation Applications," Tech. Rep., 2008. [Online]. Available: http://www.osti.gov/bridge

[142] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," *Journal of cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.

[143] H. Rhys, *Machine Learning with R, the tidyverse, and mlr*. Manning Publications, 2020. [Online]. Available: https://books.google.co.za/books?id=jRzYDwAAQBAJ

[144] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[145] B. Zu Eissen and F. Wißbrock, "On cluster validity and the information need of users," *ACTA Press*, pp. 216–221, 2003.